Relative pose determination of uncooperative known target based on extracting region of interest

Fang Yin\textsuperscript{1,2}, Wusheng Chou\textsuperscript{1,2}, Yun Wu\textsuperscript{3} and Mingjie Dong\textsuperscript{4}

Abstract
This paper deals with the problem of relative pose determination between a model-known uncooperative space target and a chaser spacecraft using three-dimensional point cloud. A novel, reliable and real-time relative pose determination framework is proposed, which is composed of a region of interest–based initial pose acquisition method and a template matching iterative closest point algorithm for tracking pose. The initial pose is obtained by the principal component analysis method which aligns the region of interest extracted from the scanning point cloud with the region of interest of the known model point cloud, and a three-dimensional convex hull is constructed to extract the region of interest fast. To improve the iterative closest point, the registration between the template point cloud generated by the pose of last frame and the point cloud of the current frame is used to replace the registration between the local scanning point cloud and the global model point cloud. In addition, the performance (stability, low cost and their robustness against noise) of the proposed initial pose acquisition method and the accuracy and reliability of the pose tracking algorithm have been demonstrated by the numerical experiments. Finally, the effectiveness of the proposed method is verified by the field experiment.

Keywords
Uncooperative target, three-dimensional point cloud, relative pose estimation, region of interest, template matching iterative closest point

Date received: 21 December 2018; accepted: 24 April 2019

Introduction
The relative pose determination in the aerospace field computes the relative position and attitude between the chaser spacecraft and the target spacecraft. As one of the critical technologies for many advanced space applications, like on-orbit servicing (OOS)\textsuperscript{1,2} and active debris removal (ADR)\textsuperscript{3,4} the relative pose determination algorithm has been successfully applied to the projects\textsuperscript{5–7} between two cooperative spacecrafts, which are equipped with auxiliary equipments such as marker and answering machine.

However, most of the spacecraft of the on-orbit service and space debris are uncooperative targets without available auxiliary markers or communication devices, which brings great challenges to the pose determination between the chaser and uncooperative targets and has been frequently concerned by the majority of researchers. The method of the pose estimation depends on the type of sensors installed on board of the chaser spacecraft. Such sensors can be monocular vision\textsuperscript{8} stereo vision\textsuperscript{9} and Light Detection and Ranging (LiDAR)\textsuperscript{10,11} Due to the robustness in the variable illumination conditions, various types of LiDAR, such as scanning LiDAR and Flash LiDAR, have become the dominant types in the research field of the pose determination of uncooperative targets.\textsuperscript{12}

In this paper, a fast and accurate initial pose estimation algorithm based on region of interest (ROI) extraction is proposed for the uncooperative spacecraft using the three-dimensional (3D) point cloud obtained from Flash LiDAR or scanning LiDAR. Due to the speed requirement, the 3D convex hull method is used to assist in extracting ROI. In addition, the PCA...
(principal component analysis) method is adopted to obtain the initial pose. In the pose tracking step, a stable strategy for judging the correctness of the initial pose is proposed first. Aiming at the phenomenon of error increase in registration between the whole model point cloud and the local scanning point cloud in pose tracking, the template matching iterative closest point (TM-ICP) method is proposed which matches the scanning point cloud and the template point cloud generated by the pose of last frame using the ICP algorithm. And the effectiveness of the proposed method is verified by extensive numerical and field experiments.

The rest of this paper is organized as follows. In the “Related work” section, the related work in recent years is described in detail. In the “ROI-based algorithm for relative pose determination” section, the details of proposed relative pose estimation method are presented. The “Pose tracking” section describes the pose tracking stage. Finally, numerical and field experiments and conclusions are shown in the “Experiments” and “Conclusion” sections, respectively.

Related work

Uncooperative targets are usually divided into two categories. One has known geometrical information about their shape and size at least, and the other is fully unknown. In the following step, we provide a brief overview of the relative pose determination for the model-known uncooperative target, which exists in large numbers in space. The process of pose estimation for the model-known uncooperative target is generally divided into the initial pose acquisition stage and the pose tracking stage.

The initial pose acquisition for the model-known uncooperative target aligns the acquired data with the target model, which is built as model point cloud, templates or complex features, corresponding to the point-based algorithm and the feature-based algorithm. A 3D binary template matching (TM) algorithm is presented in Greenspan and Jasiobedzki. It divides the 3D template points into grid elements called voxels, and quantifies the occupied voxel as a value of 1 and an empty voxel as a value of 0.

Opromolla et al. build the 3D point template either off-line or on-board by sampling the attitude parameter space with a fixed angular step and estimate the relative position by means of the centroiding approach. These algorithms take up the amount of on-board data storage or highly computational time (tens of seconds), which is not available for uncooperative spacecrafts with a high tumbling rate. To optimize computational efficiency and restrain the amount of on-board data storage, a PCA-based online TM algorithm is proposed. Similar to Greenspan and Jasiobedzki, the PCA algorithm is used to estimate the main axis of the target, which can resolve two rotational components of the target pose around the other two axes. The remaining unresolved rotation around the target main axis finds the best match by generating templates online. The performance of these techniques has been assessed by means of numerical simulations.

The methods proposed by Ruel et al. are to store the pairs of points and their associated topological information to trade space for time. The corresponding relationship between the model and the scanning point is found by looking for corresponding polygons or congruent tetrahedron composed of pairs of points, and then the initial pose parameters are estimated. Such methods attributed to the fast lookup capabilities of the hash table are available in real time.

In addition, there are some feature-based approaches that store features extracted from the model off-line, for example, the 3D semi-global descriptor OUR-CVFH (Oriented, Unique, and Repeatable Clustered Viewpoint Feature Histograms). Based on this feature, Woods and colleagues presented an OUR-CVFH algorithm for the initial pose estimation of uncooperative targets, and the high correctness and the low time complexity of the algorithm are verified by simulation experiments. However, the accuracy and stability of feature extraction from point cloud are usually sensitive to noise.

Without storing features extracted from the model, templates or other topological information of points, the Globally Optimal ICP (Go-ICP) algorithm only needs to store the model point cloud directly. It is the first globally optimal solution to Euclidean registration of two 3D points that is applied to the initial pose estimation of uncooperative targets. While the Go-ICP method takes a long time (tens of seconds at 500 points) to reduce the translation domain, Liu et al. use PCA to calculate the axis-aligned bounding box (AABB) to obtain the new domain and then use the Go-ICP algorithm to search in the 3D space.

Man-made targets often have obvious geometric shapes, such as cylinders, cones, and rectangles. Martinez et al. detect the cylinder body of the uncooperative rocket and project the point cloud onto the axis of symmetry to obtain the barycenter. The initial attitude is obtained by the relative position among the detected nozzle, side tanks, and fairing of target. Inspired by this, and taking into account the structural characteristics of satellites, this paper proposes a real-time method based on the ROI for some man-made spacecraft with large plane features, which only needs to store the model point cloud, and has higher success rate (SR), accuracy and noise resistance.

After acquiring the initial pose, it will be necessary to further refine the pose by applying the iterative closest point (ICP) algorithm which refines the pose in a continuous parameter space by minimizing an error metric iteratively or bounded Hough transform (BHT) operating in the discrete domain and effectively trades off reduced precision for increased efficiency. For the ICP method in the pose tracking stage, the error will gradually increase—one of the reasons...
for this is the wrong correspondence between the model point cloud and the local scanning point cloud. In this paper, the TM-ICP method combining the TM and ICP algorithm is presented to solve this problem effectively.

The proposed method for initial pose acquisition in this paper is a point-based method too. And the relative pose is estimated by aligning the ROI extracted from the model point cloud and the scanning LiDAR point cloud. The strategy combined with ICP and TM is adopted to track pose.

**ROI-based algorithm for relative pose determination**

The flow of proposed method for determining the relative pose between chaser and target is illustrated in Figure 1. Following the general pipeline, we also divide the process of pose determination into two steps: initial pose acquisition and pose tracking. In the initial pose acquisition step, a ROI-based algorithm was investigated. An algorithm combined with ICP and TM is used in pose tracking step for reliability and accuracy.

First, the coordinate system of the chaser spacecraft and the target spacecraft should be introduced as shown in Figure 2. $O_c - X_cY_cZ_c$ is the chaser body-fixed reference frame while $O_s - X_sY_sZ_s$ is the sensor reference frame, and the LiDAR sensor is equipped on the chaser. $O_t - X_tY_tZ_t$ is the target reference frame. The objective is to determine a transformation $T_{ct}$ (Equation (1)) which represents the relative rotation matrix $R_{ct}$, which aligns the target reference frame to the chaser reference frame, and the translation vector $t_{ct}$, which is expressed in the chaser reference frame. The transformation $T_{ct}$ is constant because of the fixed assembly between the chaser and the sensor. Therefore, only the relative transformation $T_{st}$ between the sensor and the target is required.

\[
T_{ct} = T_{cs}T_{st} \tag{1}
\]

**ROI extracting**

For most spacecrafts, there are large-scale solar arrays compared to the body size as shown in Figure 3; the length of solar arrays is significantly larger than $d_{bs}$ (the maximum distance between the point on the body of spacecraft (not on the solar arrays) and the point on the solar arrays), which is the promise of the ROI extraction in this paper. The ROI in this paper mainly refers to the points on the plane where the solar array is located. The initial pose is obtained by aligning the ROI of the scanning point cloud (SROI) with the ROI of the model point cloud (MROI).

Since the geometric model of the uncooperative spacecraft is known, the plane ROI can be directly determined based on the distance between the scanning points. The process of extracting the ROI is shown in Algorithm 1.

The plane parameters can be calculated by three non-collinear scanning points $p_1$, $p_2$ and $p_3$. To find these three points faster, a 3D convex hull of
scanning point cloud $S$ is constructed, and the searching process is performed in the vertices of the convex hull. The greatest distance between the point pairs of convex hull vertices is $L_{\text{max}} = \|p_1 - p_2\|$. If the distance $L_{\text{max}}$ is greater than the threshold $th_1$ (taking into account the measurement errors of scanning points, the distance threshold is set to $th_1$ and $th_1 > 3\sigma$), it is can be considered that the two points $p_1$ and $p_2$ lie on the plane defined by the solar arrays. Then, taking $p_1$ as the endpoint and looking for the point $p_3$ in the vertices of the convex hull after excluding $p_2$, where $\|p_1 - p_3\| > th_1$, $p_1$, $p_2$ and $p_3$ are non-collinear. From these three points $p_1$, $p_2$ and $p_3$, the plane of the solar arrays can be determined. Considering the influence of noise, the plane is not determined by $p_1$, $p_2$ and $p_3$ directly, but by the mean values of their neighbors in the scanning point cloud within radius $r$, respectively. Finally, the distance from the scanning point cloud to the plane is calculated. If the distance $d$ is less than $th_2$, the point is considered to belong to SROI (Figure 4). It should be noted that, even if the point where the main body attached to the solar arrays is incorrectly assigned to the ROI, it will not significantly affect the calculation of PCA later.

**Acquiring the initial pose**

When the SROI is extracted, the initial pose can be acquired by the flow as shown in Figure 5.

The PCA algorithm is used for acquiring the initial pose. The covariance matrix $Q_i$ is written in Equation (2), and the relative position vector $t_0$ is calculated by the centroid of the ROI in Equation (3)

$$Q_i = \frac{1}{N} \begin{bmatrix} \sum_{i=1}^{N} (x_i - SC_x)^2 & \cdots & \sum_{i=1}^{N} (x_i - SC_x) \cdot (y_i - SC_y) & \cdots & \sum_{i=1}^{N} (x_i - SC_x) \cdot (z_i - SC_z) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N} (z_i - SC_z) \cdot (x_i - SC_x) & \cdots & \sum_{i=1}^{N} (z_i - SC_z) \cdot (y_i - SC_y) & \cdots & \sum_{i=1}^{N} (z_i - SC_z) \cdot (z_i - SC_z) \end{bmatrix}$$

$$t_0 = SC - MC$$

where $SC = \{SC_x, SC_y, SC_z\}$ and $MC$ are the centroids of SROI and MROI.

**Algorithm 1. ROI Extracting**

**Input:** Scanning Point cloud, $S = \{S_i\}, i = 1, 2, ..., n$  
**Output:** ROI of the scanning Point cloud, SROI

1. Constructing the 3D convex hull of the scanning point cloud $(S_i, V_i)$ are the vertices of the convex hull, $k = 1, 2, ..., m$ ($m < n$)
2. Finding a pair of points $(p_1, p_2)$ determined the greatest distance, $p_1, p_2 \in V_i$
3. if $\|p_1 - p_2\| > th_1$ then
   4. for $i = 1$ to $k$ do
      5. $L_i = \|p_1 - V_i\|$, $V_i \neq p_2$
      6. if $L_i > th_1$ & $\alpha \sin (p_1, p_2, p_1/V_i) > 0$ then
         7. $p_3 = V_i$
         8. break for.
      9. end if
     10. end for
   11. if $p_3$ is found then
      12. for $p_j = 1, 2, 3$ do
         13. finding the neighbors $N = \{N_j\}$ of $p_j$ from $\{S_i\}$ within the radius $r$
         14. $p_j = \text{mean}(N)$
      15. end for
      16. Calculate the plane $\Pi$ with $p_1, p_2, p_3$.  
      17. for $i = 1$ to $n$ do
         18. if distance $(S_i, \Pi)$ $< th_2$ then
            19. $S_i \in$ SROI.
      20. end if
   21. end for
   22. else
      23. go to step 1 for the next frame.
      24. end if
   25. else
      26. go to step 1 for the next frame.
   27. end if
The matrix $R_S$ is composed by the eigenvectors of $Q_S$ as described in Equation (4); $V_1 = [V_{1x}, V_{1y}, V_{1z}]^T$, $V_2 = [V_{2x}, V_{2y}, V_{2z}]^T$ and $V_3 = [V_{3x}, V_{3y}, V_{3z}]^T$ are the eigenvectors corresponding to the eigenvalues $\lambda_1$, $\lambda_2$ and $\lambda_3$, respectively, and $\lambda_1 > \lambda_2 > \lambda_3$. The matrix $R_M$ can be composed following the procedure of $R_S$. Then, the relative attitude $R_0$ rotating from the target reference frame to the sensor reference frame can be calculated by the Equation (5)—both $R_S$ and $R_M$ are orthonormal

$$R_0 = R_S R_M^{-1}$$

The PCA allows estimation of the three main axes targets without ambiguity, but their directions remain undetermined. Thus, four possible solutions were generated. That is to say, there are four possibilities for $R_M$ corresponding to $R_S$. If the $Y$, $Z$ and $X$ axes of the model are the first, second and third main directions, respectively, there are four possibilities for $R_M$ as shown in Equation (6)

$$R_{M1} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, R_{M2} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, R_{M3} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix}, R_{M4} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

The four candidate rotation matrices $R_{M1}$, $R_{M2}$, $R_{M3}$ and $R_{M4}$ are determined by $R_S$ and $R_{M1}$, $R_{M2}$, $R_{M3}$ and $R_{M4}$ according to Equation (4), respectively. To validate the right rotation matrix from these four rotation matrices, the ICP algorithm is used. The four rotation matrices, along with $t_0$, are taken as the initial transformation matrices of the ICP algorithm to align the model point cloud and the scanning point cloud. The transformation matrix with the minimum score $f$ (Equation (7)) of ICP is selected as the correct initial transformation matrix

$$f = \frac{1}{n} \sum_{i=1}^{n} \|M_i - (R_S i + t)\|^2$$

where $M_i$ is the point of the model point cloud and $S_i$ is the point of the scanning point cloud.

**Pose tracking**

The correctness of the initial pose estimation directly affects the effectiveness of the pose tracking. Therefore, the correctness of the initial pose should be judged before entering the pose tracking phase. And the procedure is described in Algorithm 2. The poses $T_{st}^s$ and $T_{st}^{s-1}$ of the point clouds of two adjacent frames are calculated by the ROI-based method, respectively. The relative pose of the two frames calculated by the two initial poses is $\Delta T_{st}^{s-1} = (T_{st}^s (T_{st}^{s-1})^{-1})$. On the contrary, the ICP algorithm is used to obtain the
Algorithm 2: The procedure of the proposed pose tracking

Input: The target model point cloud $M$, the subsequence of scanning point cloud $S$, $S_i$ is the $i$-th frame scanning point cloud.

Output: the 6-DOF relative pose from model point cloud to scanning point cloud.

1: while $k > 0$ do
2: Using the ROI–based algorithm to calculate the transformation $T^k_{st}$ from $M$ to $S_{k-1}$.
3: Using the ROI–based algorithm to calculate the transformation $T^k_{st}$ from $M$ to $S_k$.
4: Using ICP algorithm to estimate the relative pose $\Delta T^k$ between $S_{k-1}$ and $S_k$.
5: $\Delta T^k = T^k_{st}^{-1}(\Delta T)^{-1}$. $\Delta T^k$.
6: Computing the Euler angles $\delta \alpha$, $\delta \beta$, and $\delta \gamma$ from $\Delta T^k$.
7: if ($\delta \alpha < \alpha_0$ and $\delta \beta < \beta_0$ and $\delta \gamma < \gamma_0$) ($\beta_0$, $\lambda_0$, and $\alpha_0$ are the thresholds) then
8: $k = k + 1$; go to step 3.
9: else
10: $T^k_{st} = T^k_{st}$; $k = k + 1$; go to step 3.
11: end if
12: TM–ICP tracking:
13: Generating the template $TM_{k-1}$ using the relative pose $T^k_{st}$.
14: Using ICP algorithm to estimate the relative pose $\Delta T^k$ between $TM_{k-1}$ and $S_k$.
15: $\Delta T^k = \Delta T^k_{st}$.
16: Calculating the 6-DOF relative pose parameters from $T^k_{st}$.
17: $k = k + 1$ and go to step 13.
18: end while

Figure 6. Flow diagram of pose tracking: (a) ICP algorithm and (b) TM-ICP algorithm.

relative transformation $\Delta T^k$ of the two frames. Due to the high precision of $\Delta T^k$, if $\Delta T^k_{st} = \Delta T^k_{st} = \Delta T^k_{st}$, the poses $T^k_{st}$ and $T^k_{st}$ are considered to be correct. The Euler angles with a 231 sequence, that is, $\delta \beta$, $\delta \gamma$, and $\delta \alpha$, are used to measure the relative pose $\Delta T^k_{st}$, and $\beta_0$, $\lambda_0$, and $\alpha_0$ are the thresholds. If $\delta \alpha < \alpha_0$, $\delta \beta < \beta_0$, and $\delta \gamma < \gamma_0$, it is considered $\Delta T^k_{st} = \Delta T^k_{st}$.

Since there is an error in the initial pose estimation, if the thresholds are set too small, even if the initial pose estimation is correct, it may always be judged to fail, resulting in the inability to enter the pose tracking normally. Therefore, the setting of the thresholds needs to be adjusted according to the accuracy of the initial pose estimation algorithm. Here, $\beta_0 = 90^\circ$, $\lambda_0 = 5^\circ$, $\alpha_0 = 90^\circ$.

The general ICP-based pose tracking process is shown in Figure 6(a). However, due to the influence of the initial pose and the registration of the local point cloud and the overall model point cloud, the error will gradually increase during the tracking process (although the error will decrease later). Generally, the registration between adjacent frames is relatively accurate and stable, and is inspired by the TM algorithm which is used for initial pose acquisition; an improved tracking strategy named TM-ICP is presented as shown in Figure 6(b). Different from the general ICP-based pose tracking process to register the scanning point cloud with the whole model point cloud, the ideal local point cloud (template) is generated by the pose of the last frame and is used to register with the current scanning point cloud to obtain the pose. The template provides a local point cloud, which has a large overlap with the current scanning point cloud, and provides a truth pose, which can make the ICP have high precision and stability. Pseudo code of TM-ICP method is shown in Algorithm 2 (TM-ICP tracking).

Experiments
In this section, numerical simulation experiments and field experiments have been performed to test the
The size of the simulation model (Figure 7(a)) is 1521 × 559 × 870 mm, and the size of the simplified mock-up (Figure 7(b)) is 2350 × 660 × 1016 mm.

A commercial desktop PC equipped with an Intel® Core i5 CPU at 2.4 GHz is used to obtain the experimental results and the proposed algorithm is implemented in C++.

Simulation experiments

ROI-based algorithm test. It is difficult to quantify the performance of the ROI-based algorithm of large pose variations in laboratory tests; thus, a simulation environment which had been described in Yin et al.15 including simulated sensor point clouds and providing the true relative pose between the uncooperative space target and the sensor, is used. The performance of the initial attitude estimation algorithm is tested in terms of both computational cost and pose estimation performance first, and then, the effectiveness of the pose tracking strategy is tested.

All the Euler angles in the experiments use the 231 convention and the ranges of variation of the Euler angles are [−180°, 180°], [−90°, 90°] and [−180°, 180°] (corresponding to β, γ and α).

Since we cannot exhaust all the attitudes in the rotation space, we sample at intervals of 10° in each rotation region of the three rotation axes, which means that there are 26,011 sets of point cloud data in each group. The number of scanning points is below 500. The anti-noise performance of the proposed method should be test too. Therefore, six groups' simulation experiments were designed in different noise conditions as shown in Table 1. The noise range [a, b] represents the range of random noise along each axis direction.

SR (Equation (8)) and loss rate (LR, Equation (9)) are defined to discriminate whether the algorithm is successful or not. A 5° angular error threshold is used to consider whether the result of pose estimation succeeds or not

\[
SR = \frac{\text{Number of successful pose estimates}}{\text{Number of satisfying the ROI-based algorithm}}
\]  

\[
LR = 1 - \frac{\text{Number of satisfying the ROI-based algorithm}}{\text{Number of sets}}
\]

The SR and LR in different noise are shown in Figure 8.

From Figure 8, it can be seen that the proposed algorithm has a high SR, even with a maximum random noise of 150 mm. The SR is more than 90% and the LR is less than 3%. As the noise increases, the SR decreases slightly, but within the noise range tested here, SR and LR are high enough for pose acquisition.

The average values of the absolute estimation error are collected in Figure 9. In the maximum noise 200 mm tested in this paper, the position error is less than 200 mm and the attitude error is less than 4°. Although the position error and the attitude error increase as the noise increases, the accuracy is still high enough for pose acquisition.

The computational time of the proposed algorithm is collected in Table 2. It can be seen that under different noises, there is no significant difference in computational time and the algorithm is faster than 5 Hz, which

### Table 1. Conditions of simulation experiments of region of interest–based algorithm.

| Range of noise (mm) | X | Y | Z |
|---------------------|---|---|---|
| Group 1             | 0 | 0 | 0 |
| Group 2             | [−10, 10] | [−10, 10] | [−10, 10] |
| Group 3             | [−10, 10] | [−10, 10] | [−50, 50] |
| Group 4             | [−10, 10] | [−10, 10] | [−100, 100] |
| Group 5             | [−10, 10] | [−10, 10] | [−150, 150] |
| Group 6             | [−10, 10] | [−10, 10] | [−200, 200] |
is fast enough for the pose tracking step of tumbling target with a high tumbling angular velocity.

**Pose tracking test.** As the spinning uncooperative spacecraft usually rotates around the maximum or minimum inertia axis, the two cases of rotating around the Y axis and Z axis (Figure 7(a)) were tested. And the TM-ICP algorithm and ICP algorithm for tracking are compared. Since the sampling frequency of LiDAR is generally higher, the distortion of the scanning point cloud is not considered in the rotation process of the target.

In test 1, the initial attitude \((\alpha, \beta, \gamma)\) is set as \((85^\circ, 0^\circ, 90^\circ)\). The target rotates around the Z axis of the model and is sampled at \(5^\circ\) interval. So the number of the simulated sensor point cloud data is 73. The Gaussian white noise with a standard deviation of 10 mm is added to \(X, Y\) and \(Z\) direction, and a Monte-Carlo simulation with 100 independent runs is performed.

The average rotation error and translation error of test 1 are given in Figures 10 and 11. The shading in the figures represents the boundary of the standard deviation of the error.

From Figures 10 and 11, it can be found that the ROI-based algorithm can provide an accurate initial attitude and a good initial position, and the TM-ICP pose tracking method has higher accuracy and stability than ICP algorithm. The rotation error is less than \(1^\circ\) and the position error is less than 10 mm.

In test 2, the initial attitude \((\alpha, \beta, \gamma)\) is set as \((0^\circ, 80^\circ, 10^\circ)\). The target rotates around the \(Y\) axis of the model and is sampled at \(5^\circ\) interval. So the number of the

---

**Table 2.** Computational time of the ROI-based method at different noises.

| Noise \((Z/\text{mm})\) | 0   | 10  | 50  | 100 | 150 | 200 |
|------------------------|-----|-----|-----|-----|-----|-----|
| Average time \((\pm \delta/\text{ms})\) | 142 ± 51 | 157 ± 59 | 152 ± 52 | 147 ± 49 | 159 ± 46 | 190 ± 80 |
simulated sensor point cloud data is 73. The Gaussian white noise with a standard deviation of 10 mm is added to X, Y and Z direction, and a Monte-Carlo simulation with 100 independent runs is performed. The rotation error and translation error of test 2 are given in Figures 12 and 13. The shading in the figures represents the boundary of the standard deviation of the error.

From Figures 12 and 13, it can be seen that the ROI-based algorithm can still provide an accurate initial attitude and a good initial position, and the TM-ICP pose tracking method has higher accuracy and stability than ICP algorithm. When the target rotates with its maximum inertia axis, it has the approximate precision and stability when rotating around the minimum inertia axis.

Field experiments

The field experiment system shown in Figure 14 consists of a simplified mock-up which is mounted on a three-axis turntable with a precision of 0.01°, a single-line scanning 3D-LiDAR system which is developed by

---

**Figure 10.** Angle error for tracking with the initial pose (85°, 0°, 90°): (a) α angle error, (b) β angle error and (c) γ angle error.

**Figure 11.** Translation error for tracking with the initial pose (85°, 0°, 90°): (a) X translation error, (b) Y translation error and (c) Z translation error.

**Figure 12.** Angle error for tracking with the initial pose (0°, 80°, 10°): (a) α angle error, (b) β angle error and (c) γ angle error.
BICE and installed on the manipulator arm, and a air-bearing guideway which floats on a near-perfectly flat floor, providing translation motion and rotation.

The target is still, and the LiDAR rotates around the target with the air-bearing guideway at a distance of 7 m from the target, and rotation speed is $2^\circ/s$. The frequency of point cloud data acquisition is set to 1 Hz, and the experimental results are shown in Figures 15 and 16.

From the Figures 15 and 16, it can be found that ROI-based method can provide the accurate enough initial pose for the next pose tracking step. In the pose tracking step, the absolute error of the attitudes is about $5^\circ$ and the absolute error of position is less than 150 mm. In addition, it should be noted that, around frame 50 and frame 159, the angle $\beta$ rotating on $Y$ axis becomes larger. By looking at the point cloud (as shown in the Figure 17 in different views), it can be found that only one plane on the side of the target model can be scanned and the measurement error of the LiDAR system is large and complex, which leads to the increase of the pose estimation error, but the error of the subsequent frame converges with the increase of the features scanned later.
Conclusion

The presented ROI-based method in this paper is both efficient, executing in less 0.2 s, and reliable, succeeding in over 90% of the almost entire space domain. And it is not limited by the type of sensors to obtain the 3D point cloud, such as scanning LiDAR, flash LiDAR or other stereo vision. For the dense point cloud, it is only used to down sample before ROI is extracted to ensure the real-time performance of the proposed algorithm. In addition, the TM-ICP method is proposed to achieve more accurate and more stable pose tracking. The results of the simulation experiments and field experiment also confirmed the effectiveness of the proposed method both in accuracy and stability.

The ROI-based method proposed in this paper is applicable to the uncooperative targets which have large-scale solar arrays (rectangle) compared to the body size. In future work, other methods of extracting regions of interest of different shapes may be considered.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship and/or publication of this article: This research was funded by the National Key R&D Program of China under grant 2017YFB1302503 and the National Natural Science Foundation of China under grant 61633002.

ORCID iD

Fang Yin https://orcid.org/0000-0001-7555-0953

References

1. NASA. On-orbit satellite servicing study project report. Technical Report, NASA, Washington, DC, 2010.
2. Rembala R and Ower C. Robotic assembly and maintenance of future space stations based on the ISS mission operations experience. Acta Astronaut 2009; 65: 912–920.
3. Bonnal C, Ruault JM and Desjean MC. Active debris removal: recent progress and current trends. Acta Astronaut 2013; 85: 51–60.
4. Thienel JK, Van Eepoel JM and Sanner RM. Accurate state estimation and tracking of a non-cooperative target vehicle. In: Proceedings of AIAA guidance, navigation, and control conference and exhibit, Keystone, CO, 21–24 August 2006.
5. Oda M. Experiences and lessons learned from the ETS-VII robot satellite. In: Proceedings of the 2000 IEEE
6. Woffinden DC and Geller DK. Navigating the road to autonomous orbital rendezvous. J Spacecr Rockets 2012; 44: 898–909.

7. Christian JA, Robinson SB, DSouza CN, et al. Cooperative relative navigation of spacecraft using flash light detection and ranging sensors. J Guid Control Dyn 2014; 37: 452–465.

8. Pesce V, Opromolla R, Sarno S, et al. Autonomous relative navigation around uncooperative spacecraft based on a single camera. Aerosp Sci Technol 2019; 94: 1070–1080.

9. Zhang LM, Zhu F, Hao YM, et al. Optimization-based non-cooperative spacecraft pose estimation using stereo cameras during proximity operations. Appl Opt 2017; 56: 4522–4531.

10. Greenspan M and Jasiobedzki P. Pose determination of a free-flying satellite. In: Proceedings of the 2002 International Conference on Imaging Science, Systems, and Technology (CISST’2002), Las Vegas, Nevada, USA, 24–27 June 2002, pp. 1–6. Boston: CSREA Press.

11. Rhodes A, Kim E, Christian JA, et al. LIDAR-based relative navigation of non-cooperative objects using point cloud descriptors. In: Proceedings of the AIAA/AAS astrodynamics specialist conference, Long Beach, CA, 12–15 September 2016.

12. Christian JA and Cryan S. A survey of LIDAR technology and its use in spacecraft relative navigation. In: Proceedings of AIAA guidance, navigation, and control (GNC) conference, Boston, MA, 19–22 August 2013.

13. Wold S, Esbensen K and Geladi P. Principal component analysis. Chemometrics Intell Lab Syst 1987; 2: 37–52.

14. Opromolla R, Fasano G, Rufino G, et al. Uncooperative pose estimation with a LIDAR-based system. Acta Astronaut 2015; 110: 287–297.

15. Yin F, Chou W, Wu Y, et al. Sparse unorganized point cloud based relative pose estimation for uncooperative space target. Sensors 2018; 18: 1009.

16. Opromolla R, Fasano G, Rufino G, et al. Pose estimation for spacecraft relative navigation using model-based algorithms. IEEE Trans Aerosp Electron Syst 2017; 53: 431–447.

17. Ruel S, Luu T, Ancill M, et al. Target localization from 3d data for on-orbit autonomous rendezvous and docking. In: Proceedings of the 2008 IEEE aerospace conference, Big Sky, MT, 1–8 March 2008, pp. 1–11. New York: IEEE.

18. Aldoma A, Tombari F, Rusu RB, et al. OUR-CVFH: oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6DOF pose estimation. In: Pizn A, Pock T, Bischof H, et al. (eds) Pattern recognition. DAGM/OAGM 2012. Lecture notes in computer science. Berlin; Heidelberg: Springer, pp. 113–122.

19. Woods JO and Christian JA. Lidar-based relative navigation with respect to non-cooperative objects. Acta Astronaut 2012; 126: 298–311.

20. Yang J, Li H and Jia Y. Go-ICP: solving 3D registration efficiently and globally optimally. In: Proceedings of the 2013 IEEE international conference on computer vision, Sydney, NSW, Australia, 3–6 December 2013, pp. 1457–1464. New York: IEEE.

21. Liu L, Zhao G and Bo Y. Point cloud based relative pose estimation of a satellite in close range. Sensors 2016; 126: 824.

22. Martinez HG, Giorgi G and Eissfeller B. Pose estimation and tracking of non-cooperative rocket bodies using Time-of-Flight cameras. Acta Astronaut 2017; 139: 165–175.

23. Besl PD and McKay ND. A method for registration of 3-D shapes. IEEE PAMI 1992; 14: 239–256.

24. Rusinkiewicz S and Levoy M. Efficient variants of the ICP algorithm. In: Proceedings third international conference on 3-D digital imaging and modeling, Quebec City, QC, Canada, 28 May–1 June 2001, pp. 145–152. New York: IEEE.

25. Zhao G, Xu S and Bo Y. LiDAR-based non-cooperative tumbling spacecraft pose tracking by fusing depth maps and point clouds. Sensors 2018; 18: 3432.

26. Greenspan M, Shang L and Jasiobedzki P. Efficient tracking with the Bounded Hough Transform. In: Proceedings of the 2004 IEEE computer society conference on computer vision & pattern recognition, Washington, DC, 27 June–2 July 2004.

27. Shang L, Jasiobedzki P and Greenspan M. Model-based tracking by classification in a tiny discrete pose space. IEEE Trans Pattern Anal Mach Intell 2017; 29: 976–989.

28. NASA 3D resources, https://nasa3d.arc.nasa.gov/ (accessed 1 November 2018).

29. Barber CB, Dobkin DP and Huhdanpaa H. The quickhull algorithm for convex hulls. ACM Trans Math Softw 1996; 22: 469–483.