

Article

Not peer-reviewed version

Rectangular Natural Feature Recognition and Pose Measurement Method for Non-Cooperative Spacecraft

[Fengxu Wang](#)*, [Wenfu Xu](#), [Lei Yan](#), Chengqing Xie, [Weihua Pu](#)

Posted Date: 3 January 2024

doi: 10.20944/preprints202401.0029.v1

Keywords: Non-cooperative targets; Parallelogram fitting; Pose measurement; Stereovision; Rectangle



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Rectangular Natural Feature Recognition and Pose Measurement Method for Non-Cooperative Spacecraft

Fengxu Wang ^{1,*} , Wenfu Xu ^{1,2}, Lei Yan ¹, Chengqing Xie ³ and Weihua Pu ³

¹ School of Mechanical Engineering and Automation, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China; lei.yan@hit.edu.cn (L.Y.)

² State Key Laboratory of Robotics and System, Harbin 150001, China

³ Shenzhen Aerospace Dongfanghong Satellite Ltd., Shenzhen 518057, China

* Correspondence: wangfengxu_hit@163.com

Abstract: Accurately estimating the pose of spacecraft is indispensable for space applications. However, such targets are generally non-cooperative, i.e., no markers are mounted on it and no parts for operation. Therefore, the detection and measurement of a non-cooperative target is very challenging. Stereovision sensors are important solutions in the near field. In this paper, a rectangular natural feature recognition and pose measurement method for non-cooperative spacecraft is proposed. Solar panels or the bodies of spacecraft are selected as detection objects and their image features are captured by stereo vision. These rectangle features are then reconstructed in 3D Cartesian space through parallelogram fitting on the image planes of two cameras. The vertexes are used to calculate the center position of the rectangle and the direction of the fixed coordinate system to solve the pose and attitude of the non-cooperative target. An experimental system is built to validate the effectiveness of the algorithm. The experimental results show that the average position measurement error of the algorithm is about 10mm and the average attitude measurement error is less than 1°. The results show that the proposed method has high accuracy and efficiency.

Keywords: non-cooperative targets; parallelogram fitting; pose measurement; Stereovision; rectangle

1. Introduction

The advancement of space technology has brought to the forefront a range of critical challenges, such as spacecraft maintenance and rescue in orbit [1], the removal of space debris [2], satellite rendezvous and docking [3], companion flight monitoring, and in-orbit capture [4]. These are urgent problems that demand effective solutions. Key to the successful execution of in-orbit missions is target recognition and pose measurement, which play pivotal roles [5]. Many of the targets in these missions are non-cooperative, lacking known structural, size, or motion information, and often without cooperative markers or communication capabilities with tracking spacecraft. Identifying and measuring non-cooperative targets poses a significant and challenging task [6,7].

Target detection can usually be divided into three stages in space [8]: long-distance (hundreds of meters to tens of kilometers), mid-distance (several kilometers to tens of meters) and close-distance (a few hundred meters to several meters). Generally, the first two stages usually use microwave radar and lidar to detect the approximate location information of the target. They have advantages in long-distance measurement, but the measurement accuracy is low and it is difficult to obtain attitude information [9,10]. For close-distance target detection, visual measurement equipment are usually used to provide the position and attitude information of targets. This article focuses on the pose measurement of non-cooperative spacecraft in short distance.

Some space programs and research departments have adopted a number of methods for close target detection. The Canadian Space Agency and NASA jointly developed triangulation LIDAR for the close-range measurement. But the point cloud matching database needs to be pre-trained [11,12]. Liu used flash laser radar to generate point cloud data, and proposed a pose tracking method based on the known satellite model [13]. Tzschichholz measured the non-cooperative targets by comparing

the 3D point cloud features observed by the ToF camera with the actual 3D models [14]. Klionovska used the ToF camera to detect the non-cooperative targets combining target 3D model [15]. Gao et al. proposed an algorithm for measuring the position and attitude of targets using a monocular camera and a laser rangefinder [16]. The ranging accuracies of the pose measurement technique based on lidar and vision are usually respectively centimeter and millimeter. In order to achieve fine work at close range, visions are often chosen as the solution. We typically want to transfer images from on-orbit operations back to the ground for preservation and analysis [17]. Therefore, satellites are usually equipped with cameras (with different optimal imaging distances and observation directions), and no additional equipment is required. It is difficult for monocular vision to measure the pose of the target without prior information. As for the multi-vision, the object information obtained has more redundancy, which can enhance robustness and measurement accuracy of the system, while the cost price is higher and the operation speed is relatively slow. Binocular visions are widely concerned by researchers because of its balance between detection accuracy and operation speed.

There are some examples of binocular stereo vision used to detect non-cooperative targets. There are two main types of target detection methods. The first one is using deep neural network models to achieve target detection and pose estimation [18,19]. Daigo proposed the AI-assisted near-field pose estimation method of spacecraft whose shape and size are completely known. This method detected the keypoints of spacecraft with machine learning techniques and solve the pose by the Efficient Perspective-n-Point (EPnP) algorithm and particle swarm optimizer (PSO) method [20]. Different deep learning approaches have more or less the following problems. Pre-training is expensive and difficult to cover a variety of targets. It requires high computing power of the onboard computer. Many researches are focused on target detection, and there are few researches on target pose solving. The second method relies on image processing techniques to identify the salient features of the target, including but not limited to triangular brackets [21], rectangular solar panels, and circular engine nozzles [22]. Xu introduced a binocular measurement approach for assessing the interface ring and engine nozzle of rockets [23], while Peng devised a virtual stereo vision measurement method reliant on the interplay of triangular and circular features, albeit with the need for manual selection of interior points [24]. Additionally, Qiao proposed a relative pose measurement technique based on point features [25], but it necessitates prior knowledge about the target. Feature-based Target measurement based on feature detection is susceptible to environment and illumination. But it has the advantages of fast computing speed and low cost.

This paper takes the method of identifying the salient features of the target. In theory, larger target features tend to yield higher recognition success rates. Therefore, this paper focuses on researching the aircraft body or rectangular solar panel, which represents the most prominent area within the target. The surface shapes of the aircraft body and the solar panels are usually rectangles. A stereo vision detection method for rectangular features is studied in this paper. The detection of rectangular features in both space and ground applications typically involves processes such as edge line feature extraction, template matching, rectangle fitting, and minimum rectangle envelope extraction. Solar panels consist of numerous small rectangular photo receptors, resulting in a multitude of straight-line features. Extracting these straight-line features is time-consuming, and accurately discerning the target lines can be challenging. Template matching demands prior knowledge of the target, so it is not suitable for the non-cooperative spacecraft detection. Regarding rectangular fitting or minimal rectangular envelope extraction, the accuracy of these methods is limited due to the distortion of rectangular features in Cartesian space when projected onto the image plane [26–28]. The error is large in fitting the contour by means of rectangular envelope.

Through the subsequent analysis, it can be found that the projection of the rectangular feature on the image plane is approximately parallelogram. Therefore, we study the parallelogram fitting method of rectangular features and apply it to non-cooperative spacecraft detection. The algorithm has following advantages: (1) The algorithm has strong adaptability and can be used to detect rectangular features that most spacecrafts have; (2) The calculation efficiency of the algorithm is high, because the

fine elements and interference elements inside the shape envelope will be gradually eliminated during operation, greatly reducing the amount of calculation; (3) The algorithm has high fitting accuracy. It combines the imaging characteristics of rectangular features. The major contributions of this paper are presented as follows: (1) The imaging characteristics of rectangles, which are the typical natural characters, in the image plane are analyzed. A parallelogram fitting method for rectangular features in the image plane is proposed; (2) The parallelogram fitting method is applied to the measurement of non-cooperative spacecraft. Natural features on non-cooperative targets such as spacecraft bodies and solar panels are detected and measured without adding additional cooperative markers.

This paper is organized as follows: Section 2 outlines the mission of detecting non-cooperative spacecraft and proposes the framework of this paper's non-cooperative spacecraft measurement method. Section 3 demonstrates the properties of rectangular features when projected onto the image plane. In the fourth section, we introduce the parallelogram fitting algorithm for rectangular features in Cartesian space. Section five presents the pose solution algorithm for non-cooperative space targets. The sixth section describes the establishment of an experimental system to evaluate our proposed method. Finally, the last section provides a summary of the paper and draws conclusions.

2. Space On-Orbit Measurement Tasks and Version Measurement Algorithm Framework

2.1. Space On-Orbit Measurement Tasks

Figure 1 shows the process of detecting and approaching a non-cooperative spacecraft during on-orbit operation. The servicing satellite detects the target satellite from long distance like hundreds of meters to tens of kilometers or predict the target satellite's orbit. The servicing satellite monitors the target at long-distance or mid-distance and reconstructs the target point clouds. In the whole space on-orbit service process, this paper focuses on the final tracking phase, that is, the Euler distance between the target and the service star is between 1m to 30m. The stereo camera is mounted on the servicing satellite. This camera is utilized to observe the non-cooperative target satellite and compute its position and orientation, which in turn is used to plan the movements of the control satellite.

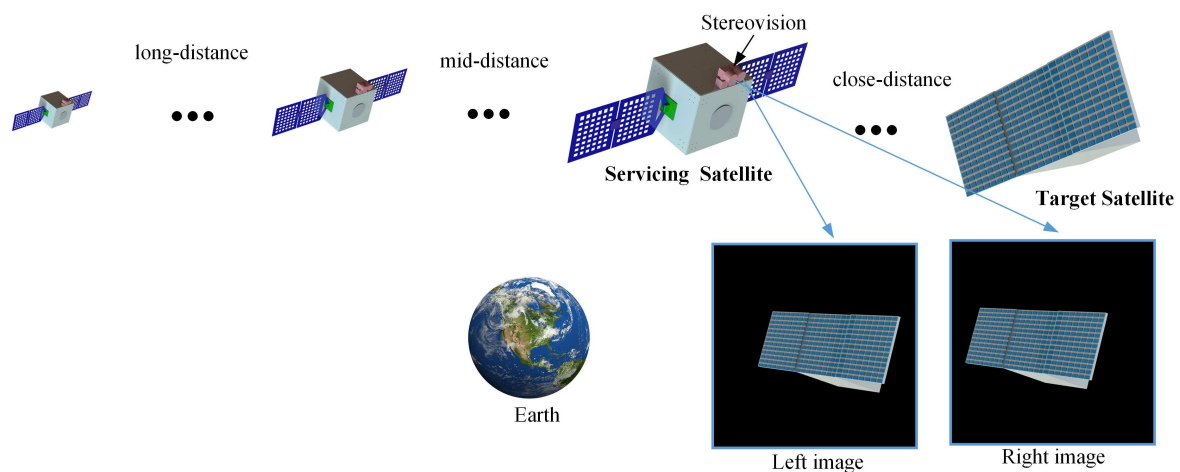


Figure 1. On orbit operation of satellite.

In the space environment, lighting conditions are both harsh and subject to rapid changes. To ensure effective detection, it is prudent to choose the most prominent target on the satellite, as depicted in Figure 2. The solar panel, with its expansive surface area and conspicuous visibility when deployed, emerges as the optimal choice for visual detection. Notably, solar panels typically exhibit a rectangular plane surface, rendering them an ideal fit for the parallelogram fitting method introduced in this paper.

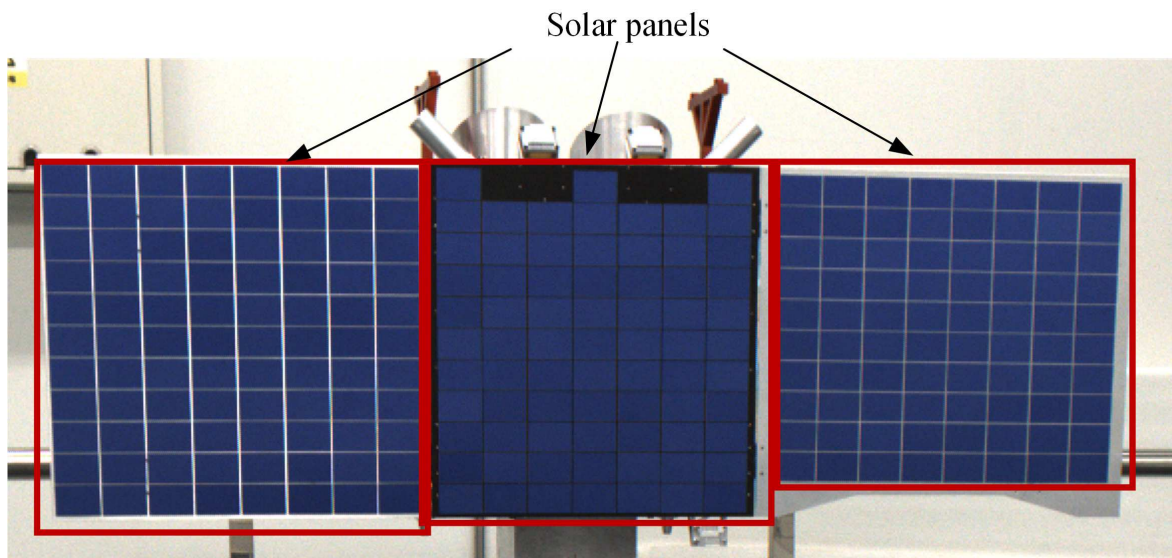


Figure 2. The target satellite.

2.2. Overview of the On-Orbit Pose Measurement Method

The process of the pose measurement algorithm for non-cooperative targets is shown in Figure 3. It comprises the following steps:

Step 1: After capturing the image of the non-cooperative target using a stereo camera, the image distortion is corrected based on pre-calibrated camera parameters.

Step 2: In the initial detection phase, search for the satellite across the entire image. To enhance algorithm speed, the region of interest (ROI) can be defined based on previous measurement results in subsequent detections.

Step 3: Convert the image to the HLS format and apply median filtering.

Step 4: Set detection criteria based on the HLS values of each pixel, extract the target region, and convert it into a binary image.

Step 5: To minimize interference from silicon wafer gaps on the extracted image's integrity and improve the extraction of the solar panel, perform morphological closing on the binary map.

Step 6: Search for closed contours in the image and fit these contours with minimum rectangular envelopes.

Step 7: Since the solar panel might have internal interference, it may be divided into multiple contours. Fit rectangles to each contour. Merge contour points based on the center point distance of each rectangle and the rectangle side length. Then, fit the minimum rectangle to the newly generated point set. Repeat this step until no further contour merging is possible.

Step 8: Select target contour points based on contour size, the area relationship between the contour and the rectangle, and the shape of the fitted rectangle.

Step 9: Utilize the parallelogram fitting algorithm proposed in this paper to fit the contour points and identify the corner points of the fitted quadrilateral.

Step 10: Solve the pose of the non-cooperative satellite in Cartesian space relative to the stereo vision system based on the corner points.

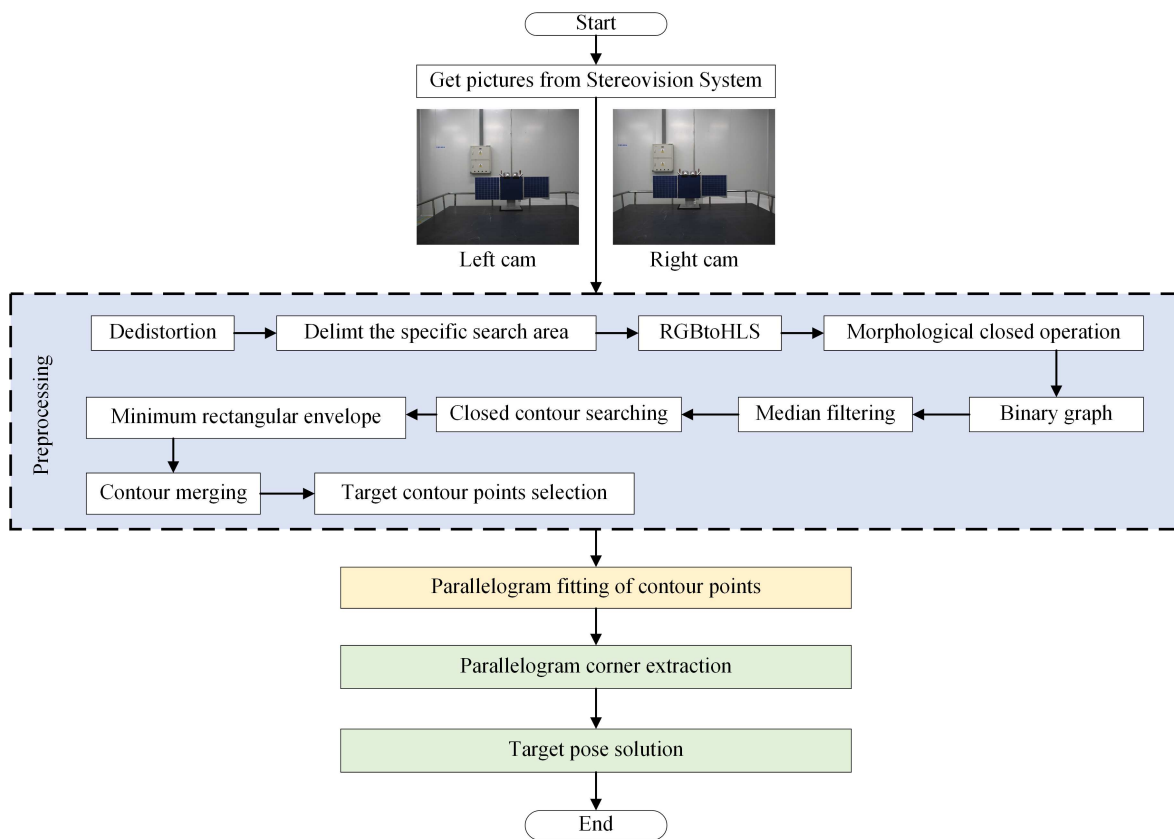


Figure 3. The process of the pose measurement algorithm for non-cooperative targets.

3. Planar Projection Properties of Rectangular Features

The majority of cameras operate on a center projection model, where features from Cartesian space are subject to distortion upon projection onto the camera's image plane. The characteristics of rectangular features when projected onto the camera image plane are illustrated as follows. The projection of a rectangle from Cartesian space onto the image plane of the camera is depicted in Figure 4. The camera coordinate system $O_c-X_cY_cZ_c$ is taken as the world coordinate system, and the coordinate of Cartesian space rectangle $P_1P_2P_3P_4$ in the camera coordinate system is (X_n, Y_n, Z_n) ($n = 1,2,3,4$). The coordinates they project into the image plane $o_c-x_cy_c$ are $p_n(x_n, y_n)$ ($n = 1,2,3,4$). According to the principle of central projection:

$$\begin{cases} x_n = f \frac{X_n}{Z_n} \\ y_n = f \frac{Y_n}{Z_n} \end{cases} \quad (1)$$

Where, f is the focal length of the camera.

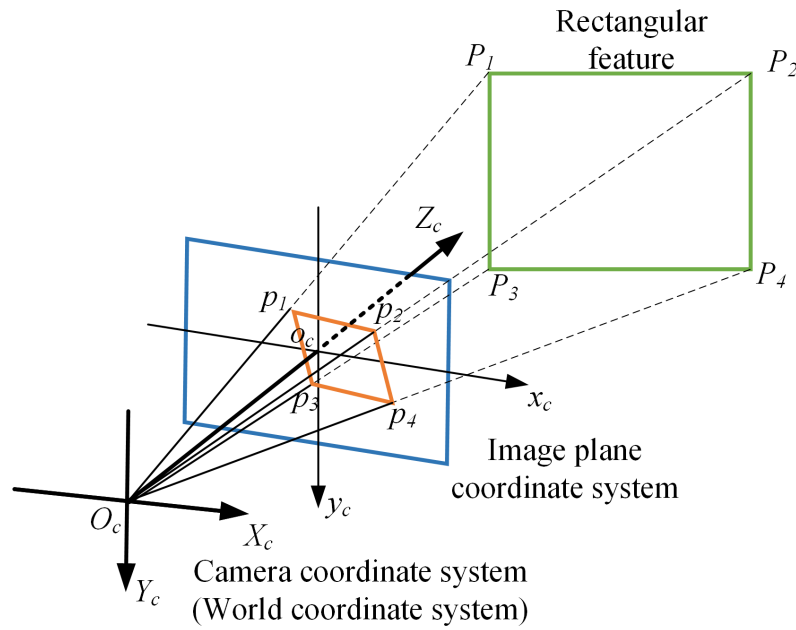


Figure 4. The image in image plane.

Rectangular features in 3D space have the following two properties when projected to 2D image plane:

Property 1: Vertical segments in the same plane in Cartesian space are not necessarily vertical in the image plane.

prove: In the camera coordinate system, $\overrightarrow{P_1P_2} \perp \overrightarrow{P_1P_3}$, so:

$$\overrightarrow{P_1P_2} \cdot \overrightarrow{P_1P_3} = (X_2 - X_1)(X_3 - X_1) + (Y_2 - Y_1)(Y_3 - Y_1) + (Z_2 - Z_1)(Z_3 - Z_1) = 0 \quad (2)$$

In the image plane:

$$\begin{aligned} \overrightarrow{p_1p_2} \cdot \overrightarrow{p_1p_3} &= (x_2 - x_1)(x_3 - x_1) + (y_2 - y_1)(y_3 - y_1) = \\ &= f^2 \left[\left(\frac{X_2}{Z_2} - \frac{X_1}{Z_1} \right) \left(\frac{X_3}{Z_3} - \frac{X_1}{Z_1} \right) + \left(\frac{Y_2}{Z_2} - \frac{Y_1}{Z_1} \right) \left(\frac{Y_3}{Z_3} - \frac{Y_1}{Z_1} \right) \right] \end{aligned} \quad (3)$$

In general, the distance between the target and the camera in the z-axis direction is much larger than the size of the solar panel, so $Z_1 \approx Z_2 \approx Z_3$, and

$$\begin{aligned} \overrightarrow{p_1p_2} \cdot \overrightarrow{p_1p_3} &\approx f^2 [(X_2 - X_1)(X_3 - X_1) + (Y_2 - Y_1)(Y_3 - Y_1)] / Z_1 \\ &\neq f^2 \overrightarrow{P_1P_2} \cdot \overrightarrow{P_1P_3} / Z_1 = 0 \end{aligned} \quad (4)$$

So $\overrightarrow{p_1p_2} \cdot \overrightarrow{p_1p_3} \neq 0$, which means $\overrightarrow{p_1p_2}$ and $\overrightarrow{p_1p_3}$ are not vertical. When $Z_1 = Z_2$ or $Z_1 = Z_3$, $\overrightarrow{p_1p_2}$ or $\overrightarrow{p_1p_3}$ is parallel to the plane of the image, and $\overrightarrow{p_1p_2}$ and $\overrightarrow{p_1p_3}$ are almost perpendicular.

Property 2: When parallel line segments in Cartesian space are far away from the image plane and the length of line segments is short, their projections in the image plane are approximately parallel.

prove: In the camera coordinate system, $\overrightarrow{P_1P_2} // \overrightarrow{P_3P_4}$ and $|\overrightarrow{P_1P_2}| = |\overrightarrow{P_3P_4}|$, so:

$$\frac{X_2 - X_1}{X_4 - X_3} = \frac{Y_2 - Y_1}{Y_4 - Y_3} = \frac{Z_2 - Z_1}{Z_4 - Z_3} = 1 \quad (5)$$

The following conditions exist for the size of solar panels in Cartesian space: $X_i - X_j, Y_i - Y_j, Z_i - Z_j \ll Z_n, Z_1 \approx Z_2 \approx Z_3 \approx Z_4$. In the image plane, the slopes of p_1p_2 and p_3p_4 are $k_{p_1k_{p_2}}$ and $k_{p_3k_{p_4}}$ respectively:

$$\begin{cases} k_{p_1k_{p_2}} = (y_2 - y_1)/(x_2 - x_1) = f(Y_2/Z_2 - Y_1/Z_1)/[f(X_2/Z_2 - X_1/Z_1)] \\ \quad = (Z_1Y_2 - Z_2Y_1)/(Z_1X_2 - Z_2X_1) \approx (Y_2 - Y_1)/(X_2 - X_1) \\ k_{p_3k_{p_4}} = (y_4 - y_3)/(x_4 - x_3) \approx (Y_4 - Y_3)/(X_4 - X_3) = (Y_2 - Y_1)/(X_2 - X_1) \end{cases} \quad (6)$$

So $k_{p_1p_2} \approx k_{p_3p_4}$, which means $p_1p_2 // p_3p_4$, similarly, $p_1p_3 // p_2p_4$.

Indeed, parallel lines in Cartesian space will converge at infinity when projected onto the image plane. However, when these three-dimensional parallel lines occupy only a few pixels in the image plane and are distant from the camera plane, they can be approximated as nearly parallel in the image plane. In the on-orbit operation, when the target is close to the camera, the relative attitude error between the target and the camera is generally very small. The proof of property 2 still holds. Furthermore, the algorithm presented in this paper includes a correction step for the fitted parallel lines in the image plane. This correction aligns the fitted lines with the actual target contour rather than maintaining their parallel nature. This adjustment ensures that the fitted lines closely adhere to the contour points, enhancing the accuracy and effectiveness of the fitting process.

4. The Parallelogram Fitting Algorithm for Rectangular Features

4.1. Parallelogram Fitting Algorithm Framework

The comprehensive procedure of the parallelogram fitting algorithm, as proposed in this paper, is delineated in Figure 5. The algorithm can be fed with input data in the form of a closed contour, a point set, or a set of contours. Here is a step-by-step breakdown of the algorithm: Step 1: Linear fitting of contour points, resulting in the centerline L of contour points based on the input point set. Step 2: Division of the point set into N_1 and N_2 on both sides of L to facilitate subsequent line fitting. Step 3: The point set for fitting parallel lines is selected and updated as N_1 and N_2 based on the distance between L and each point in these sets. Step 4: Use N_1 and N_2 to fit parallel lines l_1 and l_2 , updating N_1 and N_2 according to the distance between each point and the parallel lines. Step 5: Solve for the two groups of points at the edges of N_1 and N_2 . These two sets of points are interconnected to form a starting line for locating another set of points. Step 6: Preliminarily confirm the point set used to fit another group of parallel lines and subsequently fit those parallel lines. Step 7: Iteratively confirm the point sets used to fit the two groups of parallel lines and perform the fitting. Four sets of points and two sets of parallel lines are cyclically updated based on the distance between each point in the point sets and the lines. The distance threshold reduces as the iteration count increases to exclude points that significantly interfere with the parallel line fitting. When the elements of each point set no longer shrink after two iterations, the final point sets for fitting the four edges are derived. Step 8: To enhance the fitting precision, four lines are fitted separately using the four point sets. The intersection of these four lines yields the final fitting quadrilateral, with its four corners precisely determined. This algorithm provides a systematic approach for accurately fitting parallelograms, ensuring robust results for various input scenarios.

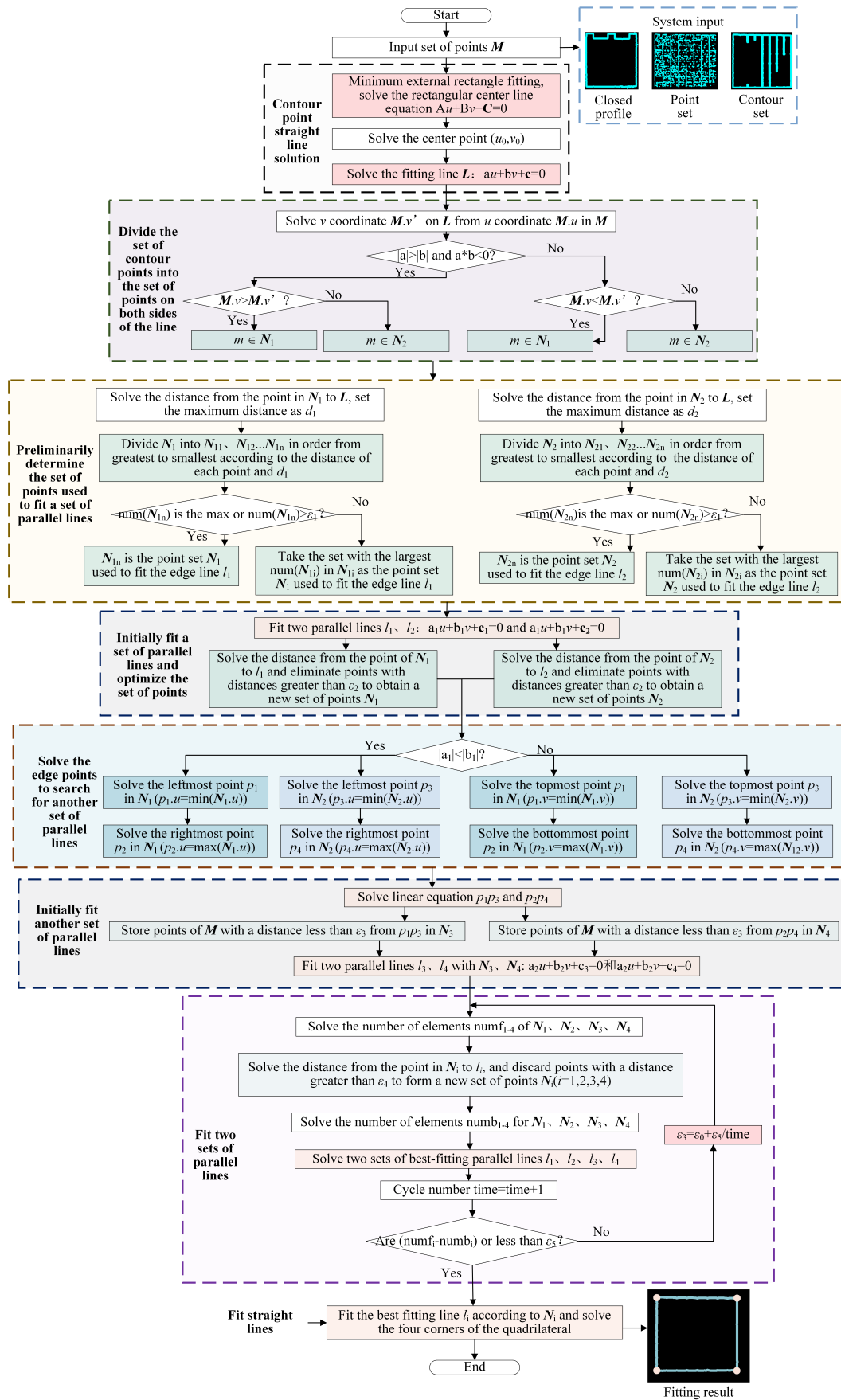


Figure 5. The framework of parallelogram fitting algorithm.

4.2. Line Fitting of Contour Points

In order to fit a line based on the input contour point set, it's necessary to first calculate the coordinates of the contour's center point. The coordinates of the center point (u_0, v_0) of the contour points M can be computed using the following formula:

$$u_0 = \frac{1}{n} \sum_{i=1}^n u_i, v_0 = \frac{1}{n} \sum_{i=1}^n v_i \quad (7)$$

L fitted by contour points satisfies the following equation:

$$\begin{cases} au + bv + c = 0 \\ a^2 + b^2 = 1 \end{cases} \quad (8)$$

The distance between contour points and L are: $au_i + bv_i + c$. To solve the center line L of the contour point is to solve a , b and c , so that the minimum value of the following formula can be obtained:

$$f = \sum (au_i + bv_i + c)^2 \quad (9)$$

Solve the above equation with the Lagrange multiplier method:

$$\begin{cases} f = \sum (au_i + bv_i + c)^2 - \lambda(a^2 + b^2 - 1) \\ \frac{\partial f}{\partial a} = 0 \quad \frac{\partial f}{\partial b} = 0 \quad \frac{\partial f}{\partial c} = 0 \quad \frac{\partial f}{\partial \lambda} = 0 \end{cases} \quad (10)$$

Simplified above formula:

$$\begin{bmatrix} \sum(x_i - \bar{x})^2 & \sum(x_i - \bar{x})(y_i - \bar{y}) \\ \sum(x_i - \bar{x})(y_i - \bar{y}) & \sum(y_i - \bar{y})^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \lambda \begin{bmatrix} a \\ b \end{bmatrix} \quad (11)$$

It is obtained by solving for the eigenvalue:

$$\lambda = \frac{[\sum(x_i - \bar{x})^2 + \sum(y_i - \bar{y})^2 - \sqrt{(\sum(x_i - \bar{x})^2 - \sum(y_i - \bar{y})^2)^2 + 4(\sum(x_i - \bar{x})(y_i - \bar{y}))^2}]}{2} \quad (12)$$

So:

$$\begin{cases} \begin{pmatrix} a \\ b \end{pmatrix} = \frac{\begin{pmatrix} \sum(x_i - \bar{x})(y_i - \bar{y}) \\ \lambda - \sum(x_i - \bar{x})^2 \end{pmatrix}}{\sqrt{(\sum(x_i - \bar{x})(y_i - \bar{y}))^2 + (\lambda - \sum(x_i - \bar{x})^2)^2}} \\ c = -a\bar{x} - b\bar{y} = \frac{-(\sum(x_i - \bar{x})(y_i - \bar{y}))\bar{x} - (\lambda - \sum(x_i - \bar{x})^2)\bar{y}}{\sqrt{(\sum(x_i - \bar{x})(y_i - \bar{y}))^2 + (\lambda - \sum(x_i - \bar{x})^2)^2}} \end{cases} \quad (13)$$

4.3. Parallelogram Fitting

In the preceding section, we successfully fitted the centerline L . We then proceeded to segment the set of contour points into two distinct sets: one on each side of L . By considering the distance between each point and the centerline, we initially identified two point sets, N_1 and N_2 , which were earmarked for fitting a pair of parallel lines, l_1 and l_2 . Following this, we selected edge points from N_1 and N_2 and used them to determine two additional point sets, N_3 and N_4 , for fitting yet another pair of parallel lines within the parallelogram. These newly determined point sets, N_3 and N_4 , were then employed to initiate the fitting of another set of parallel lines, l_3 and l_4 .

4.3.1. Solving of the points sets on both sides of the center line

Firstly, the points in the contour point set are assigned to the point sets on both sides of the line according to their position relation with L . Replace u_i in each point $m(u_i, v_i)$ in the contour point set with line $L: au + bv + c = 0$ to solve v_i' . The relationship between each point in the contour point set and L can be summarized into the following four categories, as shown in Figure 6. As shown in Figure 6 (1) and (2), the slope of L is $|-a/b| < 1$. If $v_i < v_i'$, $m \in N_1$, otherwise $m \in N_2$. As shown in Figure 6 (3), the slope of L is $-a/b \leq -1$. If $v_i < v_i'$, $m \in N_1$, otherwise $m \in N_2$. As shown in Figure 5 (4), the slope of L is $-a/b \geq 1$. If $v_i > v_i'$, $m \in N_1$, otherwise $m \in N_2$.

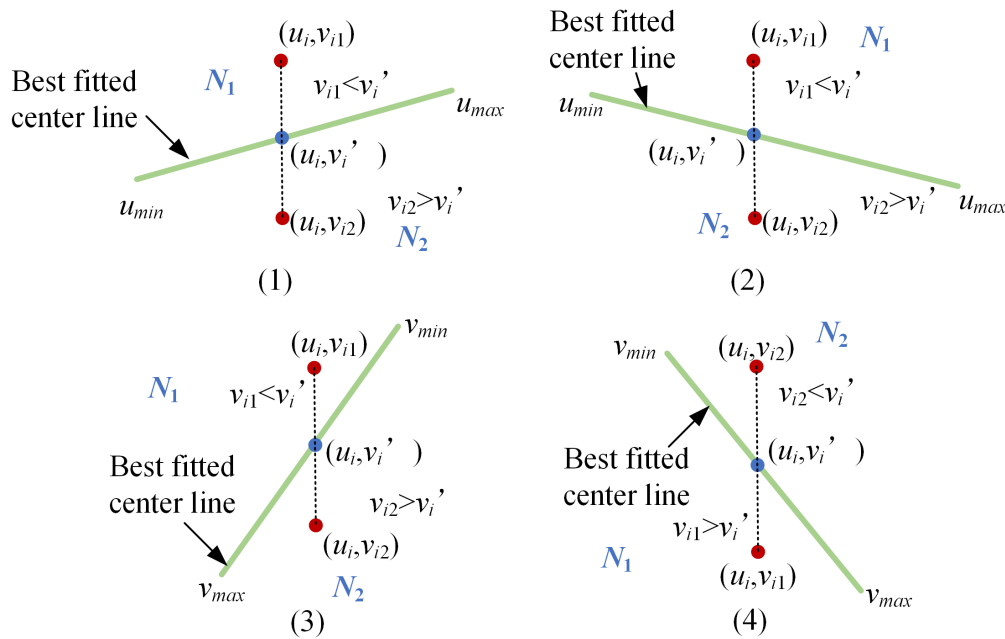


Figure 6. The framework of parallelogram fitting algorithm.

4.3.2. Initially fitting of a set of parallel lines

A set of parallel lines are initially fitted with N_1 and N_2 . Solve the farthest distance d_1 and d_2 of the points in N_1 and N_2 to L with the follow formula:

$$\begin{cases} d_1 = \max_{i=1}^{n_1} |au_i + bv_i + c| \\ d_2 = \max_{j=1}^{n_2} |au_j + bv_j + c| \end{cases} \quad (14)$$

Where, n_1 and n_2 are the number of elements of N_1 and N_2 .

We calculate the distances between the points in N_1 and N_2 with respect to the best fitting line L . These distances are classified based on their deviation from the values d_1 and d_2 . In theory, the set of edge points should be the farthest from L . Therefore, the set of points exhibiting the greatest deviation from the line should be used as the basis for fitting the parallel lines. However, if the number of elements in a point set is too small, it suggests that the set mainly comprises noise points. In such cases, we select the point set with the largest number of elements for fitting the parallel lines, which is then updated as N_1 and N_2 .

With N_1 and N_2 at hand, we proceed to fit a set of parallel lines. We utilize Equations (9) through (13) to fit the centerlines l_1 and l_2 for N_1 and N_2 , with respective slopes denoted as k_1 and k_2 . The inclination angles of these lines are represented by θ_1 and θ_2 , respectively.

$$\theta_1 = \arctan k_1 \quad \theta_2 = \arctan k_2 \quad (15)$$

To ensure that the slopes are equal, set the slope angle of parallel lines $\theta = (\theta_1 + \theta_2)/2$. Then the l_1 and l_2 are:

$$\begin{cases} \sin\theta u - \cos\theta v + c_1 = a_1 u + b_1 v + c_1 = 0 & c_1 = -a_1 \bar{u}_{N_1} - b_1 \bar{v}_{N_1} \\ \sin\theta u - \cos\theta v + c_2 = a_1 u + b_1 v + c_2 = 0 & c_2 = -a_1 \bar{u}_{N_2} - b_1 \bar{v}_{N_2} \end{cases} \quad (16)$$

Calculate the distance between points in N_1 and l_1 and the distance between points in N_2 and l_2 . The points whose distance is less than the threshold are retained and reconstructed into N_1 and N_2 .

4.3.3. Initially fitting another set of parallel lines

For fitting parallel lines, it is essential to partition the point set used for this purpose. Based on the N_1 and N_2 sets obtained in the previous section, we can determine a line using the edge points of the point set, and then search for the points near this line to form the set required for fitting parallel lines. The parallel lines, l_1 and l_2 , fitted in the previous section exhibit four possible scenarios, as illustrated in Figure 7.

When $|a_1| < |b_1|$, the target edge points are chosen based on the u coordinates of points in N_1 and N_2 , as demonstrated in Figure 7(1) and (2). The points in N_1 and N_2 with the smallest u coordinates correspond to the leftmost points, denoted as p_1 and p_3 , while those with the largest u coordinates correspond to the rightmost points, denoted as p_2 and p_4 . Conversely, when $|a_1| > |b_1|$, the selection of target edge points is based on the v coordinates of points in N_1 and N_2 , as illustrated in Figure 7(3) and (4). In this case, the points in N_1 and N_2 with the smallest v coordinates represent the topmost points, designated as p_1 and p_3 , while those with the largest v coordinates correspond to the bottommost points, denoted as p_2 and p_4 .

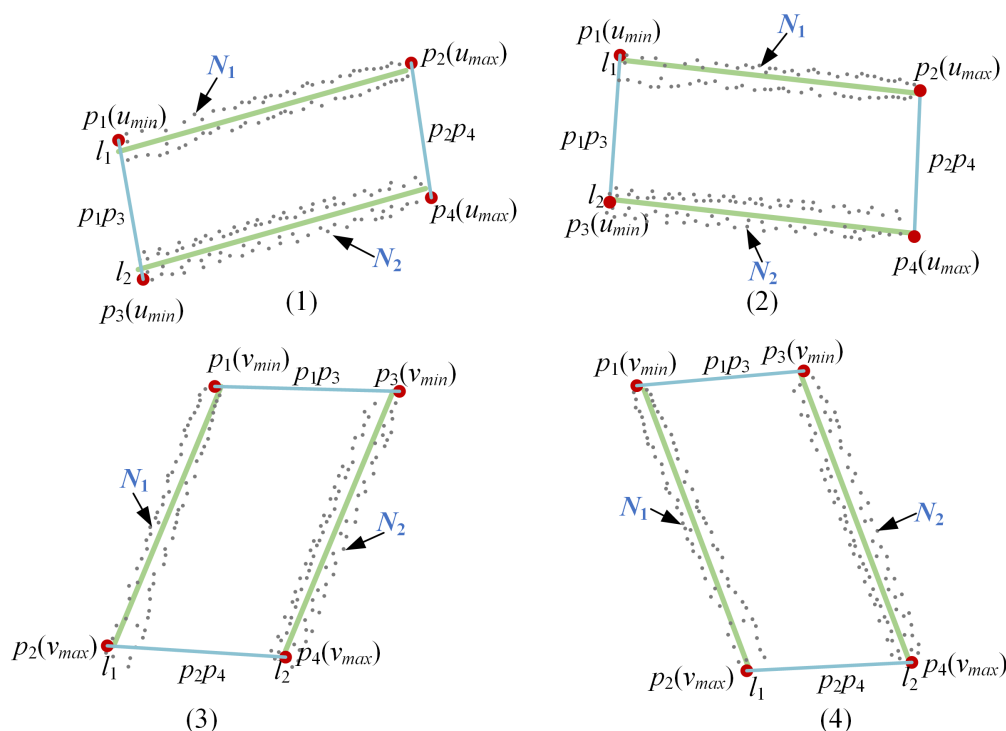


Figure 7. Edge line solution based on N_1 and N_2 .

Connect p_1p_3 and p_2p_4 to form two lines. Take p_1p_3 as an example, its equation is:

$$p_1p_3 : \frac{\frac{v_1 - v_3}{u_1 - u_3}}{\sqrt{1 + \left(\frac{v_1 - v_3}{u_1 - u_3}\right)^2}}u - \frac{1}{\sqrt{1 + \left(\frac{v_1 - v_3}{u_1 - u_3}\right)^2}}v + \frac{v_1 - \frac{v_1 - v_3}{u_1 - u_3}u_1}{\sqrt{1 + \left(\frac{v_1 - v_3}{u_1 - u_3}\right)^2}} = 0 \quad (17)$$

To proceed, we calculate the distance between each point in the contour point set M and the line segments p_1p_3 and p_2p_4 . Points whose distance is less than a predetermined threshold are selected to form the point sets N_3 and N_4 . Referring to equations (16) and (17), we then utilize N_3 and N_4 to initiate the fitting of another set of parallel lines, l_3 represented as $a_2u + b_2v + c_3 = 0$, and l_4 represented as $a_2u + b_2v + c_4 = 0$.

4.3.4. Parallelogram fitting

Following the steps described above, we have successfully obtained two sets of parallel lines, l_1 , l_2 , and l_3 , l_4 , along with four point sets N_i ($i = 1, 2, 3, 4$). During the iterative process, points in N_i that are beyond the threshold distance from l_1 are iteratively removed, and l_1 is re-fitted after each update of point set N_i . The distance threshold between points and the parallel lines decreases with each iteration. When, in both cycles, the number of elements in N_i falls below the threshold, it indicates that each point set has essentially reached its optimal configuration, and the iterative process is terminated.

Rectangles in Cartesian space only approximate parallelograms in the image plane. To further enhance the fitting accuracy, we use equations (9) through (13) to fit their optimal fitting lines, l_i , respectively. At this point, l_1 and l_2 , as well as l_3 and l_4 , need not remain parallel. These four lines intersect to form the final fitted quadrilateral. We employ the following formula to calculate the u and v coordinates of the four corners of the quadrilateral:

$$\begin{cases} u = (b_jc_j - b_jc_i)/(a_jb_j - a_jb_i) \\ v = (a_ic_j - a_ic_i)/(a_jb_i - a_ib_j) \end{cases} \quad (18)$$

5. Pose Solution Method of Non-Cooperative Target

As shown in Figure 8, stereo vision is used to detect the corner points of the satellite. The coordinates of the satellite solar panel corners extracted by the left and right cameras in the pixel plane are respectively $p_{li} = [u_{li}, v_{li}]$, $p_{ri} = [u_{ri}, v_{ri}]$ ($i = 1, 2, \dots, 12$).

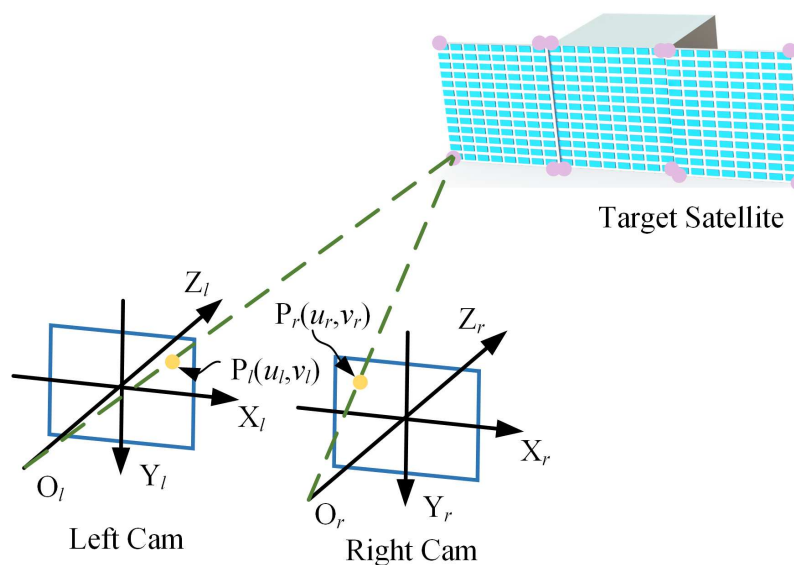


Figure 8. Satellite corners are detected through stereo vision.

The left camera coordinate system is the world coordinate system. The position P_i of each corner point of the satellite solar panel is [29]:

$$[X_i \ Y_i \ Z_i]^T = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T \mathbf{Q} \quad (19)$$

where.

$$\left\{ \begin{array}{l} \mathbf{P} = \begin{bmatrix} u_{li}m_{31}^l - m_{11}^l & u_{li}m_{32}^l - m_{12}^l & u_{li}m_{33}^l - m_{13}^l \\ v_{li}m_{31}^l - m_{21}^l & v_{li}m_{32}^l - m_{22}^l & v_{li}m_{33}^l - m_{23}^l \\ u_{ri}m_{31}^r - m_{11}^r & u_{ri}m_{32}^r - m_{12}^r & u_{ri}m_{33}^r - m_{13}^r \\ v_{ri}m_{31}^r - m_{21}^r & v_{ri}m_{32}^r - m_{22}^r & v_{ri}m_{33}^r - m_{23}^r \end{bmatrix} \\ \mathbf{Q} = [m_{14}^l - u_{li} \quad m_{34}^l m_{24}^l - v_{li} \quad m_{34}^l m_{14}^r - u_{ri} \quad m_{34}^r m_{24}^r - u_{ri} m_{34}^r] \end{array} \right. \quad (20)$$

m_{ij}^l and m_{ij}^r represent the transformation matrix parameters that relate the pixel plane coordinate system to the world coordinate system for the left and right cameras, respectively. To enhance the accuracy of stereo vision measurements, a procedure involving plane fitting to the detected solar panel corners is conducted, where the projected corner points onto the fitted plane serve as the corrected corner points. The specific steps are as follows:

The least square method is used to fit the solar panel plane. In the left camera coordinate system, the depth information Z_i of each corner point is greater than 0, and the fitting plane will not be parallel to the Z -axis of the left camera coordinate system (Because in parallel, the solar panel is a line in the image and cannot be measured). Therefore, the plane equation \mathbf{M} can be set as: $ax + by + c - z = 0$. For the fitted plane, the sum of distances from each point to the plane is minimum, that is:

$$\left\{ \begin{array}{l} \min s = \sum_{i=1}^{12} (aX_i + bY_i + c - Z_i)^2 \\ \frac{\partial s}{\partial a} = \frac{\partial s}{\partial b} = \frac{\partial s}{\partial c} = 0 \end{array} \right. \quad (21)$$

So,

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum X_i^2 & \sum X_i Y_i & \sum X_i \\ \sum X_i Y_i & \sum Y_i^2 & \sum Y_i \\ \sum X_i & \sum Y_i & 12 \end{bmatrix}^{-1} \begin{bmatrix} \sum X_i Z_i \\ \sum Y_i Z_i \\ \sum Z_i \end{bmatrix} \quad (22)$$

The equation of the plane \mathbf{M} is:

$$\left\{ \begin{array}{l} \frac{a}{m}x + \frac{b}{m}y - \frac{1}{m}z + \frac{c}{m} = Ax + By + Cz + D = 0 \\ m = \sqrt{a^2 + b^2 + 1} \end{array} \right. \quad (23)$$

The projection coordinates $P_{Mi}(X_{Mi}, Y_{Mi}, Z_{Mi})$ of each corner point of the satellite on the plane \mathbf{M} are:

$$\left\{ \begin{array}{l} X_{Mi} = X_i - At \\ Y_{Mi} = Y_i - Bt \\ Z_{Mi} = Z_i - Ct \end{array} \right. \quad (24)$$

$$t = \frac{AX_i + BY_i + CZ_i + D}{A^2 + B^2 + C^2}$$

As shown in Figure 9, with the center of the solar panel as the origin of the coordinate system, the translation matrix of the solar panel relative to the camera coordinate system is $\mathbf{T} = (P_{M3} + P_{M4} + P_{M9} + P_{M10})/4$. The solar panel coordinate system Z -axis in the camera coordinate system is: $[r_{M13}, r_{M23}, r_{M33}]^T = [A, B, C]^T$. The X -axis in the camera coordinate system is:

$$\begin{bmatrix} r_{M11} \\ r_{M12} \\ r_{M13} \end{bmatrix} = \frac{\sum_{i=1}^6 P_{M2i} - \sum_{i=1}^6 P_{M(2i-1)}}{6 \left\| \sum_{i=1}^6 P_{M2i} - \sum_{i=1}^6 P_{M(2i-1)} \right\|} \quad (25)$$

So the Y-axis in the camera coordinate system is:

$$\begin{bmatrix} r_{M12} \\ r_{M22} \\ r_{M32} \end{bmatrix} = \begin{bmatrix} r_{M13} \\ r_{M23} \\ r_{M33} \end{bmatrix} \times \begin{bmatrix} r_{M11} \\ r_{M21} \\ r_{M31} \end{bmatrix} \quad (26)$$

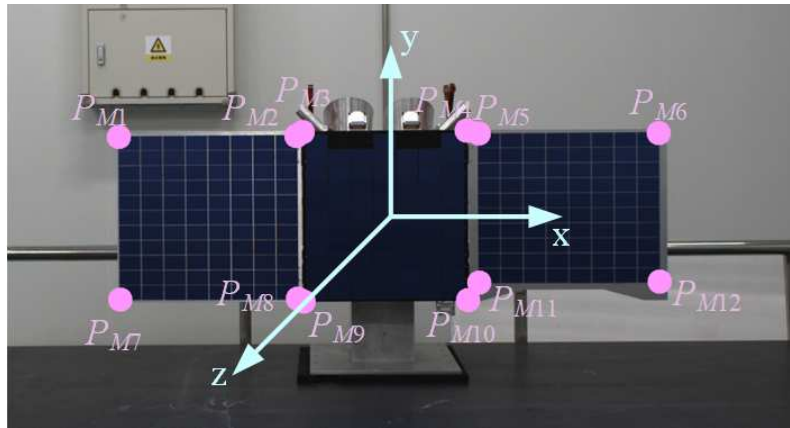


Figure 9. Coordinate system of satellite.

6. Experimental Verification

6.1. Satellite Natural Feature Recognition Experiment

To evaluate the success rate and the calculation speed of the satellite detection algorithm proposed in this paper, 100 times of tests which the distance and attitude between stereo vision and satellite varied a lot were conducted. The experimental system operated on a PC equipped with an Intel(R) i5-8500 CPU (3.0GHz) and 8.0GB of RAM. As an example, considering a monocular camera, the processing results of each step are illustrated in Figure 10. These steps successfully eliminate interference contours and points inside and outside the solar panel, while accurately extracting edge and corner points of the solar panel.

Figure 11 displays typical recognition results from these tests. It is evident that across different observation distances and angles, the algorithm presented in this paper consistently and accurately extracts the corners and contours of the satellite's solar panels. The experiments resulted in a remarkable 98.5% success rate in satellite detection, with an average detection time of only 0.14 seconds. Instances of detection failure were primarily attributed to scenarios where the solar panel plane was nearly parallel to the camera's optical axis, causing the solar panel to appear as a line in the image.

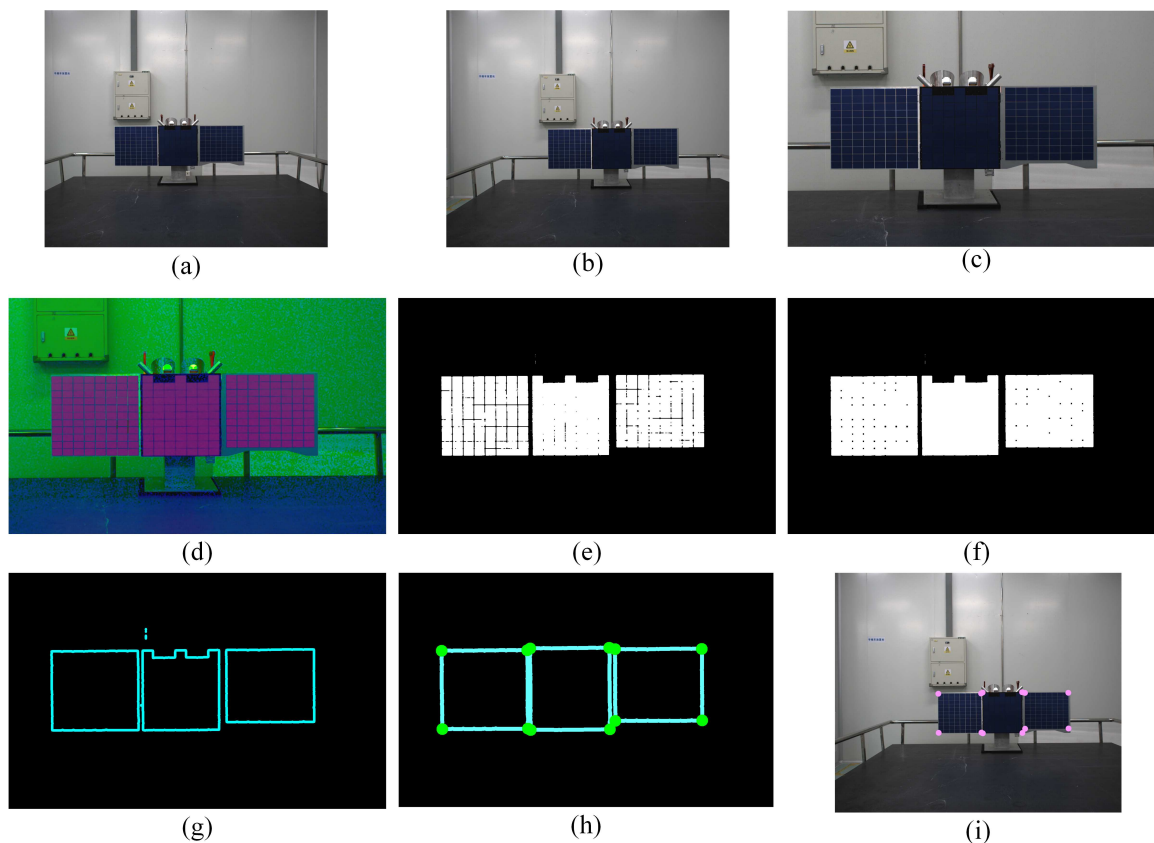


Figure 10. Non-cooperative satellite image processing effect (a)Master drawing(b)dedistortion(c)Set the ROI area(d)Median filtering(e)Binary graph(f)Morphological closed operation(g)Contour extraction(h)Parallelogram fitting(i)Solar panel corner extraction.

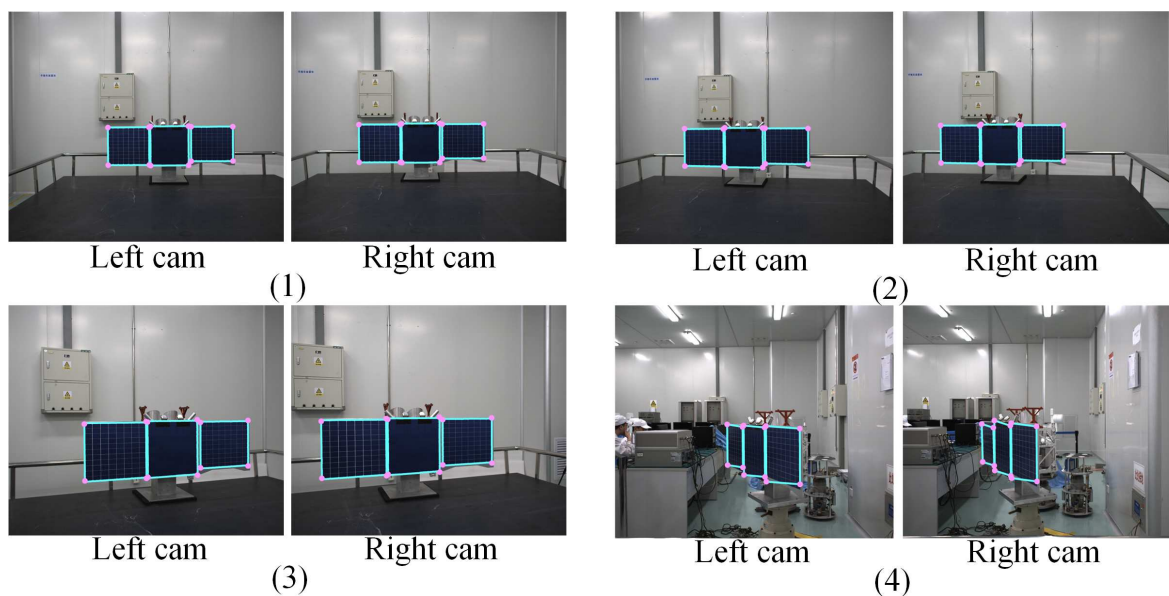


Figure 11. Satellite detection effect.

6.2. Measurement Accuracy Test of Satellite Pose

To assess the measurement accuracy of the algorithm proposed in this paper for determining the satellite's pose, we established an experimental platform, as depicted in Figure 12. This platform

featured air feet beneath both the target satellite and the service satellite, enabling them to move freely on the air platform. A stereo vision system was mounted on the service satellite, capturing images and transmitting them to a central controller that executed the algorithm presented in this paper. To ensure precise measurements, a laser tracker and calibration board were employed to calibrate the relative pose between the stereo vision system and the laser tracker, as well as to measure the pose between the satellite and the camera. The satellite's poses were measured using stereo vision. These measurements were then compared with the data obtained from the laser tracker to evaluate the measurement accuracy of the algorithm proposed in this paper. The service satellite was equipped with an LIDAR, which can synchronously measure the relative position between the target satellite and the service satellite.

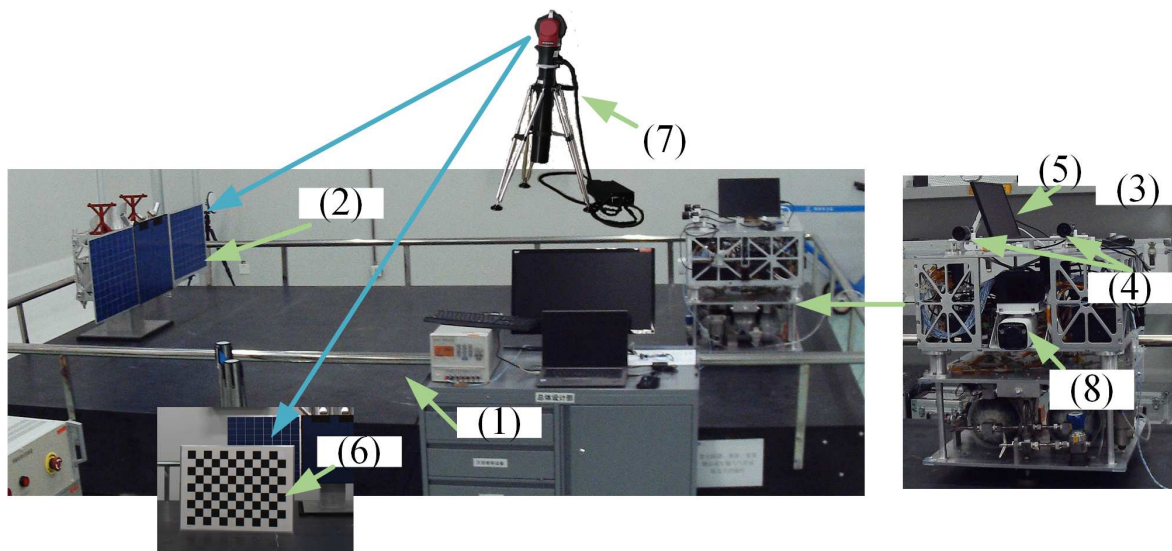


Figure 12. Experimental platform for position measurement ((1) Air float platform (2) Target satellite (3) Service satellite (4) Stereo vision (5) Central controller (6) Calibration board (7) Laser tracker (8) LIDAR).

A comprehensive set of 20 experiments was conducted, and the resulting pose measurement errors are depicted in Figure 13. Figure 13(a) provides insight into the position measurement errors and the root mean square (MSE) of the position errors in the x , y , and z directions of the satellite, while Figure 13(b) displays the attitude measurement errors and the root mean square of the attitude errors in the x , y , and z directions of the satellite. The results indicate that the measurements of pose errors remain consistently stable. The position measurement errors in the x and y directions are less than 7mm, while the position measurement error in the z direction is less than 12mm. Furthermore, the three-dimensional attitude errors are all under 1.2° , demonstrating a high level of accuracy in attitude measurement. The average measurement errors for satellite pose are summarized in Table 1, where the average position measurement error is 10.412mm and the average attitude measurement error is 0.661° . These results underscore the precision and reliability of the proposed algorithm for satellite pose measurement.

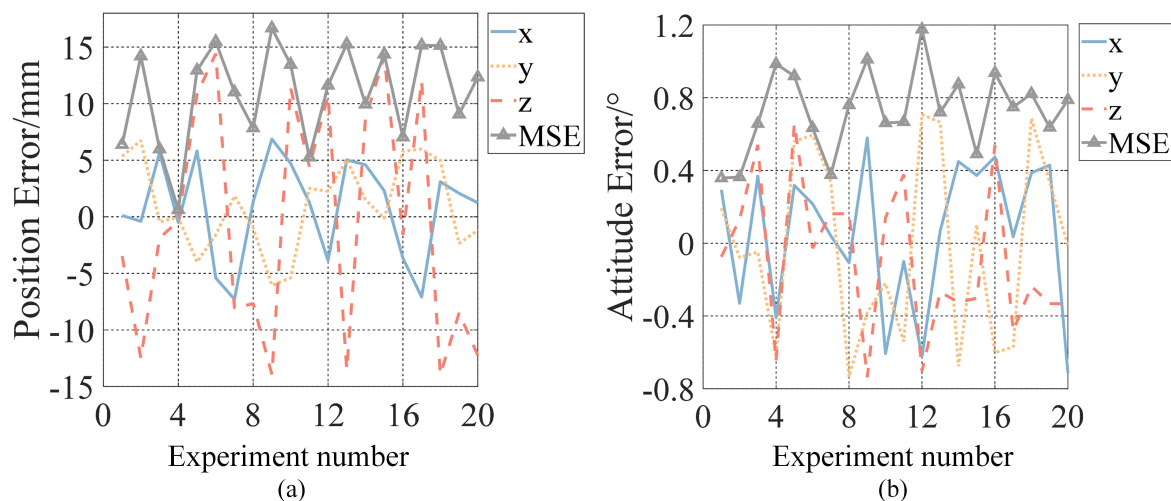


Figure 13. Pose measurement accuracy (a) Position measurement error (b) Attitude measurement error.

Table 1. Pose accuracy of satellite.

Pose errors	X	Y	Z	Root mean square
Position errors(mm)	3.617	3.213	9.219	10.412
Attitude errors(°)	0.346	0.433	0.359	0.661

The performance of various non-cooperative spacecraft detection methods can be evaluated through experimental testing and existing literature data. The detection results are shown in Table 2. In order to measure the detection success rate and detection speed of various methods, 100 experiments were carried out. As for this paper's method, the successful rate of satellite detection was 98.5%, with an average detection time of only 0.14 seconds. In comparison to existing methods, including stereo vision-based outer rectangle detection, edge line detection, and LiDAR-based detection, the proposed method stands out. The success rate of stereo vision-based detection for solar panel edge was notably lower at only 67%. This reduced success rate can be attributed to the presence of numerous straight-line features in the image due to the solar panel's composition of small rectangles, making it challenging to extract the target straight lines accurately. Overall, the method presented in this paper demonstrates superior performance in terms of speed and successful rate, surpassing all other methods except the stereo vision-based outer rectangle detection in the image plane, making it a compelling choice for satellite detection applications. As for the method of detecting satellites with deep learning in [20], the success rate was 96%. Because the algorithms are equipped with different hardware, the running time of the algorithms was not compared. Twenty experiments were carried out to test the measurement accuracy of various algorithms from different relative distance and relative attitude measurements of satellites. The test results show that the proposed method is highly accurate. External rectangle detection methods exhibit generally low accuracy, with fitting errors increasing as the deflection angle of the rectangular feature grows. Given the presence of numerous linear features on satellites, these features can significantly interfere with the extraction of the target line, resulting in a poor success rate and low accuracy in satellite edge line detection. LiDAR, for example, installed in this experimental system shown in Figure 12, typically achieves a detection accuracy of approximately 30mm, primarily capable of determining the target's position. However, it often struggles with determining the attitude of the target and often necessitates the cooperation of visual sensors for this purpose. Methods like deep learning require substantial prior information about satellites, making them costly in terms of resource requirements. Compared with [24]'s method of detecting satellite pose with stereo vision, the detection accuracy of this paper is also significantly higher. In contrast, the parallelogram fitting method for rectangular features of non-cooperative

spacecraft in the image plane, as proposed in this paper, exhibits exceptional performance. It excels in terms of detection accuracy, speed, and algorithm robustness, making it a promising choice for non-cooperative spacecraft feature detection, especially when other methods face limitations.

Table 2. Performance of various methods.

Methods	Detection success rate(%)	Detection time(s)	Position errors/Observation distance(%)	Attitude errors(°)
This paper's method	98.5	0.14	0.17	0.661
The external rectangle fitting	98.5	0.09	0.44	2.973
Laser radar	98	0.35	0.59	-
Edge line detection	67	0.29	0.25	1.088
Point clouds reconstruction by binocular vision[24]	-	-	1.327	1.16
Deep learning and pose solving by EPnP[20]	96	-	1.33	1.16
Deep learning and pose solving by PSO[20]]	96	-	0.53	1.10

7. Conclusion

This paper introduces a novel method for recognizing rectangular natural features and measuring the pose of non-cooperative spacecraft, addressing the challenging task of detecting the poses of such spacecraft in space. The method is applicable to various objects on the spacecraft, including the spacecraft body and solar panels. It employs a parallelogram fitting approach for detecting non-cooperative targets and measuring their poses. Image features are extracted using stereo vision, and the projection of rectangular features onto the image plane is then detected and fitted into parallelograms. An experimental system was established to conduct pose measurement experiments at different distances. The proposed algorithm achieved an average position measurement error of 10.412mm and an average attitude measurement error of 0.661°. These results demonstrate the method's suitability for applications of in-orbit close-range target measurement. This paper's method offers valuable insights and reference points for the development of spatial object detection algorithms and the recognition of rectangular features in Cartesian space across various working scenarios. Future research will focus on satellite detection under different illumination conditions and visual servo control.

Author Contributions: Conceptualization, F.W. and L.Y.; methodology, F.W. and L.Y.; resources, W.X.; writing—original draft, F.W.; validation, W.P.; funding acquisition, C.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62203140), Shenzhen Peacock Team Foundation (KQTD20210811090146075), Shenzhen Outstanding youth basic research (RCJC20200714114436040), the National Natural Science Foundation of China (11802073) and Shenzhen Science and Technology Innovation Commission (JSGG20191129145212206).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Sample Availability: Not applicable.

References

1. Zhi X; Yao XS; Yu F. Position and attitude joint determination for failed satellite in space close-distance approach. *J Nanjing Univ Aeronaut Astronaut* **2013**, *45*, 583–589.

2. Flores-Abad A; Ma O; Pham K. A review of space robotics technologies for on-orbit servicing. *Progress in aerospace sciences* **2014**, *68*, 1–26.
3. Shan M; Guo J; Gill E. Review and comparison of active space debris capturing and removal methods. *Progress in aerospace sciences* **2016**, *80*, 18–32.
4. Zhao Y; Zhang F; Huang P. Impulsive super-twisting sliding mode control for space debris capturing via tethered space net robot. *IEEE Transactions on Industrial Electronics* **2019**, *67(8)*, 6874–6882.
5. Huang P; Zhang F; Cai J. PDexterous tethered space robot: Design, measurement, control, and experiment. *IEEE Transactions on Aerospace and Electronic Systems* **2017**, *53(3)*, 1452–1468.
6. Segal S; Carmi A; Gurfil P. Stereovision-based estimation of relative dynamics between noncooperative satellites: Theory and experiments. *IEEE Transactions on Control Systems Technology* **2013**, *22(2)*, 568–584.
7. Rybus T. Obstacle avoidance in space robotics: Review of major challenges and proposed solutions. *Progress in Aerospace Sciences* **2018**, *101*, 31–48.
8. Li Y; Bo Y; Zhao G. Survey of measurement of position and pose for space non-cooperative target. In Proceedings of 2015 34th Chinese Control Conference, Hangzhou, China, 2015, pp.528-536.
9. Franzese V; Hein A M. Modelling Detection Distances to Small Bodies Using Spacecraft Cameras. *Modelling* **2023**, *4(4)*, 600–610.
10. Attzs M N J; Mahendrakar T; Mahendrakar T. Comparison of Tracking-By-Detection Algorithms for Real-Time Satellite Component Tracking. *Computers and Electronics in Agriculture* **2023**.
11. C. English; G. Okouneva; P. Saint-Cyr. Real-time dynamic pose estimation systems in space lessons learned for system design and performance evaluation. *Int. J. Intell. Control Syst.* **2011**, *16(2)*, 79–96.
12. P. Huang; F. Zhang; J. Cai. Dexterous tethered space robot: Design, measurement, control, and experiment. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53(3)*, 1452–1468.
13. L. Liu; G. Zhao; Y. Bo. Point Cloud Based Relative Pose Estimation of a Satellite in Close Range. *Sensors* **2016**, *16(6)*, 824–841.
14. T. Tzschichholz; T. Boge; K. Schilling. Relative Pose Estimation of Satellites Using PMD-/ccd-sensor Data Fusion. *Acta Astronautica* **2015**, *109*, 25–33.
15. K. Klionovska; H. Benninghoff. Initial Pose Estimation Using PMD Sensor During the Rendezvous Phase in On-orbit Servicing Missions. In 27th AAS/AIAA Space Flight Mechanics Meeting, San Antonio, USA, 2017, pp.263-279.
16. X. Gao; K. Xu; H. Zhang. Position-pose measurement algorithm based on single camera and laser range-finder. *J. Sci. Instrum* **2007**, *28(8)*, 1479–1485.
17. Duan F; E. H; Bernelli Zazzera F. Observer-Based Fault-Tolerant Integrated Orbit-Attitude Control of Solarsail. In INTERNATIONAL ASTRONAUTICAL CONGRESS: IAC PROCEEDINGS, 2023, pp.1-7.
18. Kilduff T; Machuca P; Rosengren A J. Crater Detection for Cislunar Autonomous Navigation through Convolutional Neural Networks. In AAS/AIAA Astrodynamics Specialist Conference, 2023, pp.1-12.
19. Mei Y; Liao Y; Gong K. SE (3)-based Finite-time Fault-tolerant Control of Spacecraft Integrated Attitude-orbit. *Journal of System Simulation* **2023**, *35(2)*, 277–285.
20. Kobayashi D; Burton A; Frueh C. AI-Assisted Near-Field Monocular Monostatic Pose Estimation of Spacecraft. In The Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference, 2023.
21. Sharma S; Beierle C; D'Amico S. Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks. In Proceedings of 2018 IEEE Aerospace Conference, 2018; pp.1-12.
22. Peng J; Xu W; Liang B. Pose measurement and motion estimation of space non-cooperative targets based on laser radar and stereo-vision fusion. *IEEE Sensors Journal* **2018**, *19(8)*, 3008–3019.
23. Jianqing P; Wenfu X. A Pose Measurement Method of a Space Noncooperative Target Based on Maximum Outer Contour Recognition. *IEEE Transactions on Aerospace and Electronic Systems* **2019**, *56(1)*, 512–526.
24. Peng J; Xu W; Liang B. Virtual Stereo-vision Pose Measurement of Non-cooperative Space Targets for a Dual-arm Space Robot. *IEEE Transactions on Instrumentation & Measurement* **2019**, *32*, 1–13.
25. Yu F; He Z; Qiao B. Stereo-vision-based relative pose estimation for the rendezvous and docking of noncooperative satellites. *Mathematical Problems in Engineering* **2014**, *21*, 1–12.
26. Chaudhuri D; Samal A; Yu F. A simple method for fitting of bounding rectangle to closed regions. *Pattern Recognition* **2007**, *40(7)*, 1981–1989.

27. Chaudhuri D; Kushwaha N K; Sharif I. Finding best-fitted rectangle for regions using a bisection method. *Machine Vision and Applications* **2011**, 23(6), 1263–1271.
28. Yang J; Jiang Z. Rectangle fitting via quadratic programming. In Proceedings of 2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP), 2015.
29. Ayache N. Rectification of images for binocular and trinocular stereovision. In Proceedings of International Conference on Pattern Recognition, Rome, Italy, 14-17 November 1988; pp. 348-379.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.