Article

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

Fengxu Wang ${ }^{1(D}$, Wenfu Xu ${ }^{1,2(\mathbb{D}}$, Lei Yan ${ }^{1, *}$, Chengqing Xie ${ }^{3}$ and Weihua Pu ${ }^{3}$<br>1 School of Mechanical Engineering and Automation, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China; wangfengxu_hit@163.com (F.W.); wfxu@hit.edu.cn (W.X.)<br>2 State Key Laboratory of Robotics and System, Harbin 150001, China<br>3 Shenzhen Aerospace Dongfanghong Satellite Ltd., Shenzhen 518057, China; xcqhit@163.com (C.X.); puweihua@szhtdfh.com (W.P.)<br>* Correspondence: lei.yan@hit.edu.cn

Citation: Wang, F.; Xu, W.; Yan, L.; Xie, C.; Pu, W. Rectangular Natural Feature Recognition and Pose Measurement Method for NonCooperative Spacecraft. Aerospace 2024, 11, 125. https://doi.org/ 10.3390/aerospace11020125

Academic Editor: M. Reza Emami

Received: 23 December 2023
Revised: 27 January 2024
Accepted: 30 January 2024
Published: 31 January 2024


[^0]
#### 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 them, and they include 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 of spacecraft were selected as detection objects, and their image features were captured via stereo vision. These rectangle features were then reconstructed in 3D Cartesian space through parallelogram fitting on the image planes of two cameras. The vertexes of rectangle features were detected and used to solve the pose of a non-cooperative target. An experimental system was built to validate the effectiveness of the algorithm. The experimental results show that the average position measurement error of the algorithm is about 10 mm and the average attitude measurement error is less than $1^{\circ}$. The results also show that the proposed method achieves 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 inorbit capture [4]. These are urgent problems that demand effective solutions. Key to the successful execution of in-orbit missions are target recognition and pose measurements, which play pivotal roles [5]. Many of the targets in these missions are non-cooperative, lacking known structural, size, or motion information, and they often lack cooperative markers or communication capabilities with tracking spacecraft. Identifying and measuring non-cooperative targets is 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). The first two stages can use microwave RADAR 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]. There are also some organizations using the observatory data for target attitude measurement. The method is applied to the attitude determination of the spacecraft GSAT-3 [11,12]. For close-distance target detection, visual measurement equipment and LIDAR are usually used to provide the pose information of targets. This article focuses on the pose measurement of non-cooperative spacecraft in a 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 close-range measurements. But the point-cloud-matching database needs to be pre-trained [5,13]. Liu used flash LIDAR to generate point cloud data, and proposed a pose tracking method based on the known satellite model [14]. Tzschichholz measured the non-cooperative targets by comparing the 3D point cloud features observed using a ToF camera with actual 3D models [15]. Klionovska used a ToF camera to detect the non-cooperative targets combining a target 3D model [16]. Gao et al. proposed an algorithm for measuring the position and attitude of targets using a monocular camera and a laser rangefinder [17]. The ranging accuracies of the pose measurement technique based on LIDAR and vision are usually, respectively, to the centimeter and to the millimeter. In order to achieve fine work at close range, vision is often chosen as the solution. We typically want to transfer images from on-orbit operations back to the ground for preservation and analysis [18]. 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 a target without prior information or other means of measurement [19,20]. As for multi-vision, the object information obtained has more redundancy, which can enhance the robustness and measurement accuracy of the system, while the cost is higher, and the operation speed is relatively slow. Researchers are widely interested in binocular vision 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 [21,22]. Daigo proposed the AI-assisted near-field pose estimation method for spacecraft whose shape and size are completely known. This method detected the keypoints of spacecraft with machine learning techniques and determined their pose using the Efficient Perspective-n-Point (EPnP) algorithm and the particle swarm optimizer (PSO) method [23]. Different deep learning approaches face more or less the following problems. Pre-training is expensive, and it is difficult to cover a variety of targets. It requires high computing power for the onboard computer. Much research has focused on target detection, and there is little research working 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 [24], rectangular solar panels, and circular engine nozzles [25]. Xu introduced a binocular measurement approach to assessing the interface ring and engine nozzle of rockets [26]. Peng devised a virtual stereo vision measurement method reliant on the interplay of triangular and circular features, albeit with the need for the manual selection of interior points [27]. Additionally, Qiao proposed a relative pose measurement technique based on point features [28], but it necessitates prior knowledge about the target. Featurebased target measurement based on feature detection is susceptible to the environment and illumination. But it offers the advantages of a fast computing speed and a low cost.

This paper applies the method of identifying the salient features of a target. In theory, larger target features tend to yield higher recognition success rates. Therefore, this paper focuses on researching the rectangular solar panel of an aircraft, which represents the most prominent area on the target. The shapes of 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 photoreceptors, 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 [29]. Template matching demands prior knowledge of the target, so it is not suitable for non-cooperative spacecraft detection. Regarding rectangular fitting or minimal rectangular envelope extraction, the accuracy of these methods is limited due to the distor-
tion of rectangular features in Cartesian space when projected onto the image plane [30-32]. The error is large when fitting the contour by means of a rectangular envelope.

Through the subsequent analysis, it could be found that the projection of the rectangular feature on the image plane is approximately a parallelogram. Therefore, we studied the parallelogram fitting method of rectangular features and applied it to non-cooperative spacecraft detection. The algorithm offers the following advantages: (1) the algorithm has strong adaptability and can be used to detect the rectangular features that most spacecraft 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; and (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 were analyzed, and a parallelogram fitting method for rectangular features in the image plane is proposed; and (2) the parallelogram fitting method was applied to the measurement of non-cooperative spacecraft, and natural features on non-cooperative targets such as solar panels were detected and measured without adding additional cooperative markers.

This paper is organized as follows: Section 2 outlines the mission of detecting noncooperative 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 Section 4, we introduce the parallelogram fitting algorithm for rectangular features in Cartesian space. Section 5 presents the pose solution algorithm for non-cooperative space targets. Section 6 describes the establishment of an experimental system to evaluate the proposed method. Section 7 provides a summary of the paper and draws conclusions.

## 2. Space On-Orbit Measurement Tasks and Vision 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 a long distance, such as hundreds of meters to tens of kilometers, or predicts the target satellite's orbit. The servicing satellite monitors the target at a long distance or a mid-distance and reconstructs the target point clouds. Of the whole space of the 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 1 m and 30 m . The stereo camera was mounted on the servicing satellite. This camera was utilized to observe the non-cooperative target satellite and compute its position and orientation, which, in turn, were used to plan the movements of the servicing 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, emerged as the optimal choice for visual detection. Notably, solar panels typically exhibit a rectangular plane surface, rendering them ideal for the parallelogram fitting method introduced in this paper.

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


Figure 1. On-orbit operation of a satellite.


Figure 2. The target satellite.
Step 3: The image is converted to the HLS (hue, lightness, and saturation) format, and median filtering is applied.

Step 4: Detection criteria are set based on the HLS values of each pixel, the target region is extracted, and it is converted into a binary image.

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

Step 6: Closed contours are searched for in the image, and these contours are fit with minimum rectangular envelopes.

Step 7: Since the solar panel might be subject to internal interference, it may be divided into multiple contours. Rectangles are fit to each contour. The contour points are merged based on the center point distance of each rectangle and the rectangle's side length. Then, the minimum rectangle is fit to the newly generated point set. This step is repeated until no further contour merging is possible.

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

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

Step 10: The pose of the non-cooperative satellite in Cartesian space is solved 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. The features in Cartesian space are subjected 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 $O_{c}-X_{c} Y_{c} Z_{c}$ is taken as the world coordinate system, and the coordinate of Cartesian space rectangle $P_{1} P_{2} P_{3} P_{4}$ in the camera coordinate system is $\left(X_{n}, Y_{n}, Z_{n}\right)(n=1,2,3,4)$. The coordinates they project into the image plane $o_{c}-x_{c} y_{c}$ are $p_{n}\left(x_{n}, y_{n}\right)(n=1,2,3,4)$. According to the principle of central projection:

$$
\left\{\begin{array}{l}
x_{n}=f \frac{X_{n}}{Z_{n}}  \tag{1}\\
y_{n}=f \frac{Y_{n}}{Z_{n}}
\end{array}\right.
$$

where $f$ is the focal length of the camera.
Rectangular features in 3D space have the following two properties when projected onto a 2D image plane:

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

Proof. In the camera coordinate system, $\overrightarrow{P_{1} P_{2}} \perp \overrightarrow{P_{1} P_{3}}$, so:

$$
\begin{equation*}
\overrightarrow{P_{1} P_{2}} \cdot \overrightarrow{P_{1} P_{3}}=\left(X_{2}-X_{1}\right)\left(X_{3}-X_{1}\right)+\left(Y_{2}-Y_{1}\right)\left(Y_{3}-Y_{1}\right)+\left(Z_{2}-Z_{1}\right)\left(Z_{3}-Z_{1}\right)=0 \tag{2}
\end{equation*}
$$



Figure 4. The image in the image plane.
In the image plane:

$$
\begin{align*}
\overrightarrow{p_{1} p_{2}} \cdot \overrightarrow{p_{1} p_{3}}= & \left(x_{2}-x_{1}\right)\left(x_{3}-x_{1}\right)+\left(y_{2}-y_{1}\right)\left(y_{3}-y_{1}\right)= \\
& f^{2}\left[\left(\frac{X_{2}}{Z_{2}}-\frac{X_{1}}{Z_{1}}\right)\left(\frac{X_{3}}{Z_{3}}-\frac{X_{1}}{Z_{1}}\right)+\left(\frac{Y_{2}}{Z_{2}}-\frac{Y_{1}}{Z_{1}}\right)\left(\frac{Y_{3}}{Z_{3}}-\frac{Y_{1}}{Z_{1}}\right)\right] \tag{3}
\end{align*}
$$

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

$$
\begin{align*}
\overrightarrow{p_{1} p_{2}} \cdot \overrightarrow{p_{1} p_{3}} & \approx f^{2}\left[\left(X_{2}-X_{1}\right)\left(X_{3}-X_{1}\right)+\left(Y_{2}-Y_{1}\right)\left(Y_{3}-Y_{1}\right)\right] / Z_{1} \\
& \neq f^{2} \overrightarrow{P_{1} P_{2}} \cdot \overrightarrow{P_{1} P_{3}} / Z_{1}=0 \tag{4}
\end{align*}
$$

So, $\overrightarrow{p_{1} p_{2}} \cdot \overrightarrow{p_{1} p_{3}} \neq 0$, which means $\overrightarrow{p_{1} p_{2}}$ and $\overrightarrow{p_{1} p_{3}}$ are not vertical. When $Z_{1}=Z_{2}$ or $Z_{1}=Z_{3}, \overrightarrow{p_{1} p_{2}}$ or $\overrightarrow{p_{1} p_{3}}$ is parallel to the plane of the image, and $\overrightarrow{p_{1} p_{2}}$ and $\overrightarrow{p_{1} p_{3}}$ are almost perpendicular.

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

Proof. In the camera coordinate system, $\overrightarrow{P_{1} P_{2}} / / \overrightarrow{P_{3} P_{4}}$ and $\left|\overrightarrow{P_{1} P_{2}}\right|=\left|\overrightarrow{P_{3} P_{4}}\right|$, so:

$$
\begin{equation*}
\frac{X_{2}-X_{1}}{X_{4}-X_{3}}=\frac{Y_{2}-Y_{1}}{Y_{4}-Y_{3}}=\frac{Z_{2}-Z_{1}}{Z_{4}-Z_{3}}=1 \tag{5}
\end{equation*}
$$

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 $p_{1} p_{2}$ and $p_{3} p_{4}$ are $k_{p 1} k_{p 2}$ and $k_{p 3} k_{p 4}$, respectively:

$$
\left\{\begin{align*}
k_{p 1} k_{p 2} & =\left(y_{2}-y_{1}\right) /\left(x_{2}-x_{1}\right)=f\left(Y_{2} / Z_{2}-Y_{1} / Z_{1}\right) /\left[f\left(X_{2} / Z_{2}-X_{1} / Z_{1}\right)\right]  \tag{6}\\
& =\left(Z_{1} Y_{2}-Z_{2} Y_{1}\right) /\left(Z_{1} X_{2}-Z_{2} X_{1}\right) \approx\left(Y_{2}-Y_{1}\right) /\left(X_{2}-X_{1}\right) \\
k_{p 3} k_{p 4} & =\left(y_{4}-y_{3}\right) /\left(x_{4}-x_{3}\right) \approx\left(Y_{4}-Y_{3}\right) /\left(X_{4}-X_{3}\right)=\left(Y_{2}-Y_{1}\right) /\left(X_{2}-X_{1}\right)
\end{align*}\right.
$$

So, $k_{p 1 p 2} \approx k_{p 3 p 4}$, which means $p_{1} p_{2} / / p_{3} p_{4}$, and similarly, $p_{1} p_{3} / / p_{2} p_{4}$.
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. What follows 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: The 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: $\boldsymbol{N}_{1}$ and $\boldsymbol{N}_{2}$ are used to fit parallel lines $\boldsymbol{l}_{1}$ and $\boldsymbol{l}_{2}$, updating $N_{1}$ and $N_{2}$ according to the distance between each point and the parallel lines. Step 5: The two groups of points at the edges of $N_{1}$ and $N_{2}$ are solved for. These two sets of points are interconnected to form a starting line for locating another set of points. Step 6: The point set used to fit another group of parallel lines and subsequently fit those parallel lines is preliminarily confirmed. Step 7: The point sets used to fit the two groups of parallel lines and perform the fitting are iteratively confirmed. 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 to accurately fitting parallelograms, ensuring robust results for various input scenarios.

### 4.2. Line Fitting of Contour Points

In order to fit a line based on the input contour point set, it is necessary to first calculate the coordinates of the contour's center point. The coordinates of the center point $\left(u_{0}, v_{0}\right)$ of the contour point $M$ can be computed using the following formula:

$$
\begin{equation*}
u_{0}=\frac{1}{n} \sum_{i=1}^{n} u_{i}, v_{0}=\frac{1}{n} \sum_{i=1}^{n} v_{i} \tag{7}
\end{equation*}
$$

$L$ fitted by contour points satisfies the following equation:

$$
\left\{\begin{array}{l}
a u+b v+c=0  \tag{8}\\
a^{2}+b^{2}=1
\end{array}\right.
$$

The distance between the contour points and $L$ is $a u_{i}+b v_{i}+c$. 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:

$$
\begin{equation*}
f=\sum\left(a u_{i}+b v_{i}+c\right)^{2} \tag{9}
\end{equation*}
$$



Figure 5. The framework of the parallelogram fitting algorithm.

The above equation is solved with the Lagrange multiplier method:

$$
\left\{\begin{array}{l}
f=\sum\left(a u_{i}+b v_{i}+c\right)^{2}-\lambda\left(a^{2}+b^{2}-1\right)  \tag{10}\\
\frac{\partial f}{\partial a}=0 \quad \frac{\partial f}{\partial b}=0 \quad \frac{\partial f}{\partial c}=0 \quad \frac{\partial f}{\partial \lambda}=0
\end{array}\right.
$$

The above formula is simplified as follows:

$$
\left[\begin{array}{cc}
\sum\left(x_{i}-\bar{x}\right)^{2} & \sum\left(x_{i}-\bar{x}\right)\left(y_{i}-\bar{y}\right)  \tag{11}\\
\sum\left(x_{i}-\bar{x}\right)\left(y_{i}-\bar{y}\right) & \sum\left(y_{i}-\bar{y}\right)^{2}
\end{array}\right]\left[\begin{array}{l}
a \\
b
\end{array}\right]=\lambda\left[\begin{array}{l}
a \\
b
\end{array}\right]
$$

It is obtained by solving for the eigenvalue:

$$
\begin{equation*}
\lambda=\frac{\left[\sum\left(x_{i}-\bar{x}\right)^{2}+\sum\left(y_{i}-\bar{y}\right)^{2}-\sqrt{\left(\sum\left(x_{i}-\bar{x}\right)^{2}-\sum\left(y_{i}-\bar{y}\right)^{2}\right)^{2}+4\left(\sum\left(x_{i}-\bar{x}\right)\left(y_{i}-\bar{y}\right)\right)^{2}}\right]}{2} \tag{12}
\end{equation*}
$$

So:

$$
\left\{\begin{array}{l}
\binom{a}{b}=\frac{\binom{\sum\left(x_{i}-\bar{x}\right)\left(y_{i}-\bar{y}\right)}{\lambda-\sum\left(x_{i}-\bar{x}\right)^{2}}}{\sqrt{\left(\sum\left(x_{i}-\bar{x}\right)\left(y_{i}-\bar{y}\right)\right)^{2}+\left(\lambda-\sum\left(x_{i}-\bar{x}\right)^{2}\right)^{2}}}  \tag{13}\\
c=-a \bar{x}-b \bar{y}=\frac{-\left(\sum\left(x_{i}-\bar{x}\right)\left(y_{i}-\bar{y}\right)\right) \bar{x}-\left(\lambda-\sum\left(x_{i}-\bar{x}\right)^{2}\right) \bar{y}}{\sqrt{\left(\sum\left(x_{i}-\bar{x}\right)\left(y_{i}-\bar{y}\right)\right)^{2}+\left(\lambda-\sum\left(x_{i}-\bar{x}\right)^{2}\right)^{2}}}
\end{array}\right.
$$

### 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 $\boldsymbol{L}$. 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 we 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, $\boldsymbol{N}_{3}$ and $\boldsymbol{N}_{4}$, 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 $L . u_{i}$ in each point $m\left(u_{i}, v_{i}\right)$ in the contour point set is replaced with line $L: a u+b v+c=0$ to solve $v_{i}^{\prime}$. 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 $v_{i}<v_{i}^{\prime}, m \in N_{1}$; otherwise, $m \in N_{2}$. As shown in Figure 6 (3), the slope of $L$ is $-a / b \leqslant-1$. If $v_{i}<v_{i}^{\prime}, m \in N_{1}$; otherwise, $m \in \boldsymbol{N}_{2}$. As shown in Figure 5 (4), the slope of $L$ is $-a / b \geqslant 1$. If $v_{i}>v_{i}^{\prime}, m \in N_{1}$; otherwise, $m \in \mathbf{N}_{2}$.

### 4.3.2. Initial Fitting of a Set of Parallel Lines

A set of parallel lines is initially fitted with $N_{1}$ and $N_{2}$. The farthest-distance $d_{1}$ and $d_{2}$ of the points in $N_{1}$ and $N_{2}$ to $L$ is solved with the follow formula:

$$
\left\{\begin{array}{l}
d_{1}=\operatorname{miax}_{i=1}^{n_{1}}\left|a u_{i}+b v_{i}+c\right|  \tag{14}\\
d_{2}=\max _{j=1}^{m_{2}}\left|a u_{j}+b v_{j}+c\right|
\end{array}\right.
$$

where $n_{1}$ and $n_{2}$ are the number of elements of $\boldsymbol{N}_{1}$ and $\boldsymbol{N}_{2}$.

(1)

(3)

(2)

(4)

Figure 6. The framework of the parallelogram fitting algorithm.
We calculated the distances between the points in $N_{1}$ and $N_{2}$ with respect to the best fitting line $\boldsymbol{L}$. These distances were 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 selected the point set with the largest number of elements for fitting the parallel lines, which was then updated as $N_{1}$ and $N_{2}$.

With $N_{1}$ and $N_{2}$ at hand, we proceeded to fit a set of parallel lines. We utilized Equations (9) through (13) to fit the centerlines $\boldsymbol{l}_{1}$ and $\boldsymbol{l}_{2}$ for $\boldsymbol{N}_{1}$ and $\boldsymbol{N}_{2}$, with respective slopes denoted as $k_{1}$ and $k_{2}$. The inclination angles of these lines are represented with $\theta_{1}$ and $\theta_{2}$, respectively.

$$
\begin{equation*}
\theta_{1}=\operatorname{arctank}_{1} \quad \theta_{2}=\operatorname{arctank}_{2} \tag{15}
\end{equation*}
$$

To ensure that the slopes are equal, the slope angle of parallel lines is set to $\theta=$ $\left(\theta_{1}+\theta_{2}\right) / 2$. Then, the $\boldsymbol{l}_{1}$ and $\boldsymbol{l}_{2}$ are:

$$
\begin{cases}\sin \theta u-\cos \theta v+c_{1}=a_{1} u+=b_{1} v+c_{1}=0 & c_{1}=-a_{1} \bar{u}_{N_{1}}-b_{1} \bar{v}_{N_{1}}  \tag{16}\\ \sin \theta u-\cos \theta v+c_{2}=a_{1} u+=b_{1} v+c_{2}=0 & c_{2}=-a_{1} \bar{u}_{N_{2}}-b_{1} \bar{v}_{N_{2}}\end{cases}
$$

The distance between points in $N_{1}$ and $\boldsymbol{l}_{1}$ and the distance between points in $N_{2}$ and $\boldsymbol{l}_{2}$ are calculated. 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

To fit 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 could 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 $\left|a_{1}\right|<\left|b_{1}\right|$, 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 $\left|a_{1}\right|>\left|b_{1}\right|$, 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 $p_{4}$.

(3)

(2)

(4)

Figure 7. Edge line solution based on $N_{1}$ and $N_{2}$.
Then, $p_{1} p_{3}$ and $p_{2} p_{4}$ are connected to form two lines. Take $p_{1} p_{3}$ as an example. Its equation is as follows:

$$
\begin{equation*}
p_{1} p_{3}: \frac{\frac{v_{1}-v_{3}}{u_{1}-u_{3}}}{\sqrt{1+\left(\frac{v_{1}-v_{3}}{u_{1}-u_{3}}\right)^{2}}} u-\frac{1}{\sqrt{1+\left(\frac{v_{1}-v_{3}}{u_{1}-u_{3}}\right)^{2}}} v+\frac{v_{1}-\frac{v_{1}-v_{3}}{u_{1}-u_{3}} u_{1}}{\sqrt{1+\left(\frac{v_{1}-v_{3}}{u_{1}-u_{3}}\right)^{2}}}=0 \tag{17}
\end{equation*}
$$

To proceed, we calculated the distance between each point in the contour point set $\boldsymbol{M}$ and the line segments $p_{1} p_{3}$ and $p_{2} p_{4}$. Points whose distance is less than a predetermined threshold were selected to form the point sets $N_{3}$ and $N_{4}$. Referring to Equations (16) and (17), we then utilized $N_{3}$ and $N_{4}$ to initiate the fitting of another set of parallel lines, $l_{3}$ represented as $a_{2} u+b_{2} v+c_{3}=0$, and $l_{4}$ represented as $a_{2} u+b_{2} v+c_{4}=0$.

### 4.3.4. Parallelogram Fitting

Following the steps described above, we successfully obtained two sets of parallel lines, $\boldsymbol{l}_{1}$ and $\boldsymbol{l}_{2}$, as well as $\boldsymbol{l}_{3}$ and $\boldsymbol{l}_{4}$, along with four point sets $\boldsymbol{N}_{i}(i=1,2,3,4)$. 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 used Equations (9) through (13) to fit their optimal fitting lines, $\boldsymbol{l}_{\boldsymbol{i}}$, respectively. At this point, $\boldsymbol{l}_{1}$ and $\boldsymbol{l}_{2}$, as well as $\boldsymbol{l}_{3}$ and $\boldsymbol{l}_{4}$, need
not remain parallel. These four lines intersect to form the final fitted quadrilateral. We employed the following formula to calculate the $u$ and $v$ coordinates of the four corners of the quadrilateral:

$$
\left\{\begin{array}{l}
u=\left(b_{i} c_{j}-b_{j} c_{i}\right) /\left(a_{i} b_{j}-a_{j} b_{i}\right)  \tag{18}\\
v=\left(a_{i} c_{j}-a_{j} c_{i}\right) /\left(a_{j} b_{i}-a_{i} b_{j}\right)
\end{array}\right.
$$

## 5. Pose Solution Method for a 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 via the left and right cameras in the pixel plane are, respectively, $p_{l i}=\left[u_{l i}, v_{l i]}, p_{r i}=\left[u_{r i}, v_{r i}\right](i=1,2 \ldots 12)\right.$.


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 [33]:

$$
\begin{equation*}
\left[X_{i} Y_{i} Z_{i}\right]^{T}=\left(\boldsymbol{P}^{T} \boldsymbol{P}\right)^{-1} \boldsymbol{P}^{T} \boldsymbol{Q} \tag{19}
\end{equation*}
$$

where

$$
\left\{\begin{array}{l}
\boldsymbol{P}=\left[\begin{array}{lll}
u_{l i} m_{31}^{l}-m_{11}^{l} & u_{l i} m_{32}^{l}-m_{12}^{l} & u_{l i} m_{33}^{l}-m_{13}^{l} \\
v_{l i} m_{31}^{l}-m_{21}^{l} & v_{l i} m_{32}^{l}-m_{22}^{l} & v_{l i} m_{33}^{l}-m_{23}^{l} \\
u_{r i} m_{31}^{r}-m_{11}^{r} & u_{r i} m_{32}^{r}-m_{12}^{r} & u_{r i} m_{33}^{r}-m_{13}^{r} \\
v_{r i} m_{31}^{r}-m_{21}^{r} & v_{r i} m_{32}^{r}-m_{22}^{r} & v_{r i} m_{33}^{r}-m_{23}^{r}
\end{array}\right]  \tag{20}\\
\boldsymbol{Q}=\left[\begin{array}{llll}
m_{14}^{l}-u_{l i} & m_{34}^{l} m_{24}^{l}-v_{l i} & m_{34}^{l} m_{14}^{r}-u_{r i} & m_{34}^{r} m_{24}^{r}-u_{r i} m_{34}^{r}
\end{array}\right]
\end{array}\right.
$$

$m_{i j}^{l}$ and $m_{i j}^{r}$ 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, in which 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 $M$ can be set as $a x+b y+c-z=0$. For the fitted plane, the sum of the distances from each point to the plane is minimal, that is:

$$
\left\{\begin{array}{l}
\min s=\sum_{i=1}^{12}\left(a X_{i}+b Y_{i}+c-Z_{i}\right)^{2}  \tag{21}\\
\frac{\partial s}{\partial a}=\frac{\partial s}{\partial b}=\frac{\partial s}{\partial c}=0
\end{array}\right.
$$

So,

$$
\left[\begin{array}{l}
a  \tag{22}\\
b \\
c
\end{array}\right]=\left[\begin{array}{ccc}
\sum X_{i}^{2} & \sum X_{i} Y_{i} & \sum X_{i} \\
\sum X_{i} Y_{i} & \sum Y_{i}^{2} & \sum Y_{i} \\
\sum X_{i} & \sum Y_{i} & 12
\end{array}\right]^{-1}\left[\begin{array}{c}
\sum X_{i} Z_{i} \\
\sum Y_{i} Z_{i} \\
\sum Z_{i}
\end{array}\right]
$$

The equation for the plane $M$ is as follows:

$$
\left\{\begin{array}{l}
\frac{a}{m} x+\frac{b}{m} y-\frac{1}{m} z+\frac{c}{m}=A x+B y+C z+D=0  \tag{23}\\
m=\sqrt{a^{2}+b^{2}+1}
\end{array}\right.
$$

The projection coordinates $P_{M i}\left(X_{M i}, Y_{M i}, Z_{M i}\right)$ of each corner point of the satellite on the plane $M$ are as follows:

$$
\begin{align*}
& \left\{\begin{array}{l}
X_{M i}=X_{i}-A t \\
Y_{M i}=Y_{i}-B t \\
Z_{M i}=Z_{i}-C t
\end{array}\right.  \tag{24}\\
& t=\frac{A X_{i}+B Y_{i}+C Z_{i}+D}{A^{2}+B^{2}+C^{2}}
\end{align*}
$$

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 $T=\left(P_{M 3}+P_{M 4}+P_{M 9}+P_{M 10}\right) / 4$. The solar panel coordinate system's Z-axis in the camera coordinate system is $\left[r_{M 13}, r_{M 23}, r_{M 33}\right]^{T}=[A, B, C]^{T}$. The X-axis in the camera coordinate system is as follows:

$$
\left[\begin{array}{l}
r_{M 11}  \tag{25}\\
r_{M 12} \\
r_{M 13}
\end{array}\right]=\frac{\sum_{i=1}^{6} P_{M 2 i}-\sum_{i=1}^{6} P_{M(2 i-1)}}{6\left\|\sum_{i=1}^{6} P_{M 2 i}-\sum_{i=1}^{6} P_{M(2 i-1)}\right\|}
$$

So, the Y -axis in the camera coordinate system is as follows:

$$
\left[\begin{array}{l}
r_{M 12}  \tag{26}\\
r_{M 22} \\
r_{M 32}
\end{array}\right]=\left[\begin{array}{l}
r_{M 13} \\
r_{M 23} \\
r_{M 33}
\end{array}\right] \times\left[\begin{array}{l}
r_{M 11} \\
r_{M 21} \\
r_{M 31}
\end{array}\right]
$$



Figure 9. Coordinate system of the satellite.

## 6. Experimental Verification

### 6.1. Satellite Natural Feature Recognition Experiment

The satellite model adopted in this paper is shown in Figure 9. In order to observe a satellite with a binocular camera, it is necessary to ensure that the target is fully imaged in the binocular camera at the same time. As shown in Figure 10, the satellite length $l$ is 1490 mm , the distance between the left and the right camera is about 291.5 mm , the field of view (FOV) is $56.3^{\circ}$, and the resolution is $2448 \times 2048$. As shown in the following formula, in order to ensure that the target can be fully imaged in both cameras at the same time, the distance between the camera and the target is at least 1668.2 mm .

$$
\begin{equation*}
L \geq(l / 2+d / 2) / \tan (F O V / 2)=1668.2 \mathrm{~mm} \tag{27}
\end{equation*}
$$

The experiment system was designed to evaluate the success rate and the calculation speed of the satellite detection algorithm proposed in this paper. The experimental system operated on a PC equipped with an $\operatorname{Intel}(\mathrm{R})$ i5-8500 CPU ( 3.0 GHz ) and 8.0 GB of RAM. As an example, considering a monocular camera, the processing results of each step are illustrated in Figure 11. These steps successfully eliminate interference contours and points inside and outside the solar panel while accurately extracting the edge and corner points of the solar panel. We changed the relative pose between the camera and the satellite and repeated the experiment 100 times to test the detection success and the detection time of the proposed method and other comparison methods. The results are shown in Table 3.

Figure 12 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 extracted 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 in which 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. Analysis of experimental scene layout requirements.


Figure 11. Non-cooperative satellite image processing effect. (a) Master drawing. (b) Dedistortion.
(c) Setting 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 12. Satellite detection effect.

### 6.2. Measurement Accuracy Test of Satellite Poses

To assess the pose measurement accuracy of the algorithm proposed in this paper, we established an experimental platform, as depicted in Figure 13. There were 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. Other methods used as control group were also tested using this system, including the external rectangle fitting method and edge line detection method. 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 stereo vision. The laser tracker was a tool used to evaluate the accuracy of this paper's algorithm. The service satellite was equipped with LIDAR, which could synchronously measure the relative position between the target satellite and the service satellite. It could be used as a control group for the method of this paper.


Figure 13. 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.

The measurement accuracy of the proposed algorithm was tested under different relative poses of the camera and the satellite. To ensure that the target could be fully imaged in the binocular camera at the same time, the observation distances included 2.5 m , 4 m , and 6 m , and the observation angles of each observation distance were divided into (1) $0^{\circ}, 0^{\circ}$, and $0^{\circ}$, (2) $0^{\circ}, 10^{\circ}$, and $0^{\circ}$, and (3) $0^{\circ},-10^{\circ}$, and $0^{\circ}$. Three tests were conducted for each group of experiments, and the average errors were calculated. The experimental results are shown in Table 1. The results show that the measurement accuracy of this method is high, and the measurement error is less affected by distance and orientation. The average measurement errors for satellite pose are summarized in Table 2. The position errors in the $x$ - and $y$-axes were less than 4 mm , and the position error in the z -axis was less than 8 mm . The average position measurement error was 9.019 mm . The attitude measurement errors of rotations about the $x-, y$-, and $z$-axes were all less than $0.5^{\circ}$. (In this paper, the attitude is defined as the Euler Angle rotating in the order of the z -axis, y -axis, and x-axis.) The average attitude measurement error was $0.571^{\circ}$. These results underscore the precision and reliability of the proposed algorithm for satellite pose measurements.

The performance of various non-cooperative spacecraft detection methods could be evaluated through experimental testing and the existing literature's data. The comparison results are shown in Table 3. For this paper's method, the successful rate of satellite detection was $98.5 \%$. 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 low success rate of solar panel edge detection can be attributed to the presence of numerous straight-line features in the image. The solar panels are composed
of small rectangles, making it challenging to extract the target straight lines accurately. Overall, the method presented in this paper demonstrated superior performance in terms of its speed and success rate, surpassing all other methods except stereo-vision-based outer rectangle detection, making it a compelling choice for satellite detection applications. The results of the measurement accuracy test show that the proposed method is highly accurate. Outer rectangle detection methods exhibit generally low accuracy, with fitting errors increasing as the deflection angle of the rectangular feature grows. The numerous linear features on satellites can significantly interfere with the extraction of the target line, resulting in a poor success rate and low accuracy in satellite edge line detection. The LIDAR installed in this experimental system, shown in Figure 13, typically achieved detection accuracy of approximately 30 mm , and it was mainly capable of determining the target's position. However, it often struggled with determining the attitude of the target, and it often necessitates the cooperation of visual sensors for this purpose. Methods like deep learning require substantial prior information about satellites, making them costly in resource requirements. Compared with the method of [27] for detecting satellite poses with stereo vision, the detection accuracy of this paper is also significantly higher. The parallelogram fitting method for the rectangular features of non-cooperative spacecraft in the image plane, as proposed in this paper, 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 1. Pose measurement accuracy.

| Distance | Orientation: $0^{\circ}, 0^{\circ}, 0^{\circ}$ |  | Orientation: $0^{\circ}, 10^{\circ}, 0^{\circ}$ |  | Orientation: $0^{\circ},-10^{\circ}, \mathbf{0}^{\circ}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Position Error (mm) | Attitude Error ${ }^{\circ}{ }^{\circ}$ ) | Position Error (mm) | Attitude Error $\left({ }^{\circ}\right)$ | Position Error (mm) | Attitude <br> Error ( ${ }^{\circ}$ ) |
| 2.5 m | 7.513 | 0.497 | 7.732 | 0.449 | 7.162 | 0.503 |
| 4 m | 8.174 | 0.421 | 9.852 | 0.575 | 8.813 | 0.518 |
| 6 m | 9.363 | 0.716 | 11.591 | 0.763 | 10.972 | 0.694 |

Table 2. Pose accuracy of the satellite.

| Pose Errors | $\mathbf{X}$ | $\mathbf{Y}$ | $\mathbf{Z}$ | Root Mean Square |
| :---: | :---: | :---: | :---: | :---: |
| Position errors $(\mathrm{mm})$ | 2.326 | 3.899 | 7.793 | 9.019 |
| Attitude errors $\left({ }^{\circ}\right)$ | 0.170 | 0.469 | 0.278 | 0.571 |

Table 3. Performance of various methods.

| Method | Detection Success <br> Rate (\%) | Detection Time (s) | Position Error (mm) | Position <br> Error/Observation <br> Distance (\%) | Attitude Error ( ${ }^{\circ}$ ) |
| :---: | :---: | :---: | :---: | :---: | :---: |

## 7. Conclusions

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 spacecraft in space. The method employs the solar panels on almost all spacecraft as its detection objects. It employs a parallelogram fitting approach to 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 test the performance of this paper's method. The proposed algorithm achieved an average position measurement error of 9.019 mm and an average attitude measurement error of $0.571^{\circ}$.

These results demonstrate the method's suitability for applications of in-orbit, close-range target measurements. This paper's method offers reference points for the development of spatial object detection algorithms and the recognition of rectangular features 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), the Shenzhen Peacock Team Foundation (KQTD20210811090146075), Shenzhen Outstanding Youth Basic Research (RCJC20200714114436040), the National Natural Science Foundation of China (11802073), and the Shenzhen Science and Technology Innovation Commission (JSGG20191129145212206).

Institutional Review Board Statement: Not applicable.
Informed Consent Statement: Not applicable.
Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: Chengqing Xie and Weihua Pu were employed by the company "Shenzhen Aerospace Dongfanghong Satellite Ltd.". The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

1. Zhi, X.; Yao, X.S.; 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. Prog. Aerosp. Sci. 2014, 68, 1-26. [CrossRef]
3. Shan, M.; Guo, J.; Gill, E. Review and comparison of active space debris capturing and removal methods. Prog. Aerosp. Sci. 2016, 80, 18-32. [CrossRef]
4. Zhao, Y.; Zhang, F.; Huang, P. Impulsive super-twisting sliding mode control for space debris capturing via tethered space net robot. IEEE Trans. Ind. Electron. 2019, 67, 6874-6882. [CrossRef]
5. Huang, P.; Zhang, F.; Cai, J.; Wang, D.; Meng, Z.; Guo, J. Dexterous tethered space robot: Design, measurement, control, and experiment. IEEE Trans. Aerosp. Electron. Syst. 2017, 53, 1452-1468. [CrossRef]
6. Segal, S.; Carmi, A.; Gurfil, P. Stereovision-based estimation of relative dynamics between noncooperative satellites: Theory and experiments. IEEE Trans. Control Syst. Technol. 2013, 22, 568-584. [CrossRef]
7. Rybus, T. Obstacle avoidance in space robotics: Review of major challenges and proposed solutions. Prog. Aerosp. Sci. 2018, 101, 31-48. [CrossRef]
8. Li, Y.; Bo, Y.; Zhao, G. Survey of measurement of position and pose for space non-cooperative target. In Proceedings of the 2015 34th Chinese Control Conference, Hangzhou, China, 28-30 July 2015; pp. 528-536.
9. Franzese, V.; Hein, A.M. Modelling Detection Distances to Small Bodies Using Spacecraft Cameras. Modelling 2023, 4, 600-610. [CrossRef]
10. Attzs, M.N.J.; Mahendrakar, T.; Mahendrakar, T. Comparison of Tracking-By-Detection Algorithms for Real-Time Satellite Component Tracking. Comput. Electron. Agric. 2023. Available online: https:/ /digitalcommons.usu.edu/smallsat/2023/all2023/ 225/ (accessed on 22 December 2023).
11. Piergentili, F.; Santoni, F.; Seitzer, P. Attitude determination of orbiting objects from light curve measurements. IEEE Trans. Aerosp. Electron. Syst. 2017, 53, 81-90. [CrossRef]
12. Piattoni, J.; Ceruti, A.; Piergentili, F. Automated image analysis for space debris identification and astrometric measurements. Acta Astronaut. 2014, 103, 176-184. [CrossRef]
13. English, C.; Okouneva, G.; Saint-Cyr, P. Real-time dynamic pose estimation systems in space lessons learned for system design and performance evaluation. Int. J. Intell. Control Syst. 2011, 16, 79-96.
14. Liu, L.; Zhao, G.; Bo, Y. Point Cloud Based Relative Pose Estimation of a Satellite in Close Range. Sensors 2016, 16, 824-841. [CrossRef]
15. Tzschichholz, T.; Boge, T.; Schilling, K. Relative Pose Estimation of Satellites Using PMD-/ccd-sensor Data Fusion. Acta Astronaut. 2015, 109, 25-33. [CrossRef]
16. Klionovska, K.; Benninghoff, H. Initial Pose Estimation Using PMD Sensor During the Rendezvous Phase in On-orbit Servicing Missions. In Proceedings of the 27th AAS/AIAA Space Flight Mechanics Meeting, San Antonio, USA, 5-9 February 2017; pp. 263-279.
17. Gao, X.; Xu, K.; Zhang, H. Position-pose measurement algorithm based on single camera and laser range-finder. J. Sci. Instrum. 2007, 28, 1479-1485.
18. Duan, F.; Xie, H.; Bernelli, Zazzera, F. Observer-Based Fault-Tolerant Integrated Orbit-Attitude Control of Solarsail. In Proceedings of the International Astronautical Congress: IAC Proceedings, Baku, Azerbaijan, 2-6 October 2023; pp. 1-7.
19. Volpe R.; Circi C.; Sabatini M. GNC architecture for an optimal rendezvous to an uncooperative tumbling target using passive monocular camera. Acta Astronaut. 2022, 196, 380-393. [CrossRef]
20. Bechini, M.; Lavagna, M.; Lunghi, P. Dataset generation and validation for spacecraft pose estimation via monocular images processing. Acta Astronaut. 2023, 204, 358-369. [CrossRef]
21. Kilduff, T.; Machuca, P.; Rosengren, A.J. Crater Detection for Cislunar Autonomous Navigation through Convolutional Neural Networks. In Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, Big Sky, MT, USA, 13-17 August 2023; pp. 1-12.
22. Mei, Y.; Liao, Y.; Gong, K. SE (3)-based Finite-time Fault-tolerant Control of Spacecraft Integrated Attitude-orbit. J. Syst. Simul. 2023, 35, 277-285.
23. Kobayashi, D.; Burton, A.; Frueh, C. AI-Assisted Near-Field Monocular Monostatic Pose Estimation of Spacecraft. In Proceedings of the The Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference, Wailea, HI, USA, 19-22 September 2023.
24. Sharma, S.; Beierle, C.; D'Amico, S. Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks. In Proceedings of the 2018 IEEE Aerospace Conference, Big Sky, MT, USA, 3-10 March 2018; pp. 1-12.
25. 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 Sens. J. 2018, 19, 3008-3019. [CrossRef]
26. Peng, J.; Xu, W.; Yan, L.; Pan, E.; Liang, B.; Wu, A.G. A Pose Measurement Method of a Space Non-cooperative Target Based on Maximum Outer Contour Recognition. IEEE Trans. Aerosp. Electron. Syst. 2019, 56, 512-526. [CrossRef]
27. Peng, J.; Xu, W.; Liang, B. Virtual Stereo-vision Pose Measurement of Non-cooperative Space Targets for a Dual-arm Space Robot. IEEE Trans. Instrum. Meas. 2019, 32, 1-13.
28. Yu, F.; He, Z.; Qiao, B. Stereo-vision-based relative pose estimation for the rendezvous and docking of noncooperative satellites. Math. Probl. Eng. 2014, 21, 1-12. [CrossRef]
29. Xu, W.; Yan, P.; Wang, F. Vision-based simultaneous measurement of manipulator configuration and target pose for an intelligent cable-driven robot. Mech. Syst. Signal Process. 2022, 165, 108347. [CrossRef]
30. Chaudhuri, D.; Samal, A.; Yu, F. A simple method for fitting of bounding rectangle to closed regions. Pattern Recognit. 2007, 40, 1981-1989. [CrossRef]
31. Chaudhuri, D.; Kushwaha, N.K.; Sharif, I. Finding best-fitted rectangle for regions using a bisection method. Mach. Vis. Appl. 2011, 23, 1263-1271. [CrossRef]
32. Yang, J.; Jiang, Z. Rectangle fitting via quadratic programming. In Proceedings of the 2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP), Xiamen, China, 19-21 October 2015.
33. Ayache, N. Rectification of images for binocular and trinocular stereovision. In Proceedings of the 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.


[^0]:    Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).

