Hindawi
International Journal of Aerospace Engineering
Volume 2017, Article ID 3162349, 11 pages
https://doi.org/10.1155/2017/3162349

Research Article

Autonomous Rendezvous and Docking with Nonfull Field of View for Tethered Space Robot

Panfeng Huang,[1,2] Lu Chen,[1,2] Bin Zhang,[1,2] Zhongjie Meng,[1,2] and Zhengxiong Liu[1,2]

[1] *National Key Laboratory of Aerospace Flight Dynamics, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China*
[2] *Research Center for Intelligent Robotics, School of Astronautics, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China*

Correspondence should be addressed to Panfeng Huang; pfhuang@nwpu.edu.cn

Received 14 October 2016; Revised 10 January 2017; Accepted 16 January 2017; Published 13 February 2017

Academic Editor: Christian Circi

In the ultra-close approaching phase of tethered space robot, a highly stable self-attitude control is essential. However, due to the field of view limitation of cameras, typical point features are difficult to extract, where commonly adopted position-based visual servoing cannot be valid anymore. To provide robot's relative position and attitude with the target, we propose a monocular visual servoing control method using only the edge lines of satellite brackets. Firstly, real time detection of edge lines is achieved based on image gradient and region growing. Then, we build an edge line based model to estimate the relative position and attitude between the robot and the target. Finally, we design a visual servoing controller combined with PD controller. Experimental results demonstrate that our algorithm can extract edge lines stably and adjust the robot's attitude to satisfy the grasping requirements.

1. Introduction

The increasing number of space debris has been a serious threat for the safety of space activities. Although many debris removal methods have already been proposed [1–3], such as the electrodynamic tether [4] and foam method [5], the majority of them are limitedly examined on ground and far from being practical. In practice, the traditional "space platform + multi-DOF manipulator arm" system has been the preferred solution; however, it has several disadvantages, such as small operating range, complicated control operation, and being only applicable for cooperative target capturing, whose attitude and movement are stable. Therefore, developing on-orbit service technologies, especially for noncooperative targets, is critical. In order to overcome the problems caused by manipulator, Huang et al. [6, 7] proposed the tethered space robot (TSR) system for noncooperative target capturing and detumbling [8, 9]. TSR is a new type of space robot, which comprises a platform, space tether, and an operational robot, as shown in Figure 1. The operational robot is released by the platform through the space tether to approach the target and accomplish an on-orbit task. Compared with the manipulator arm, TSR has two main merits: a large operational radius (as long as 1,000 m) and the enhanced flexibility.

The flow of providing an on-orbit service can be divided into three sequential stages [10]:

(a) The platform distinguishes the target from space environment and flies towards it gradually from a far distance.

(b) The platform flies around the target and detects corresponding regions to identify a suitable position to release the operational robot. Then, the operational robot will be launched out and freely fly towards the frontal target. Meanwhile, it keeps detecting the suitable grasping region of the target to guide end-effector in precise manipulation.

(c) The operational robot grasps the target once reaching the appointed position (usually less than 0.15 m) and eliminates the tumbling of the robot-target combination with its own propellants. Then it provides on-orbit services, such as injecting propellants and dragging the target into a graveyard orbit.

In the third stage, the operational robot needs to capture the target spacecraft through the docking system (such as the docking ring) in order to provide on-orbit services. But the docking system usually works for cooperative targets

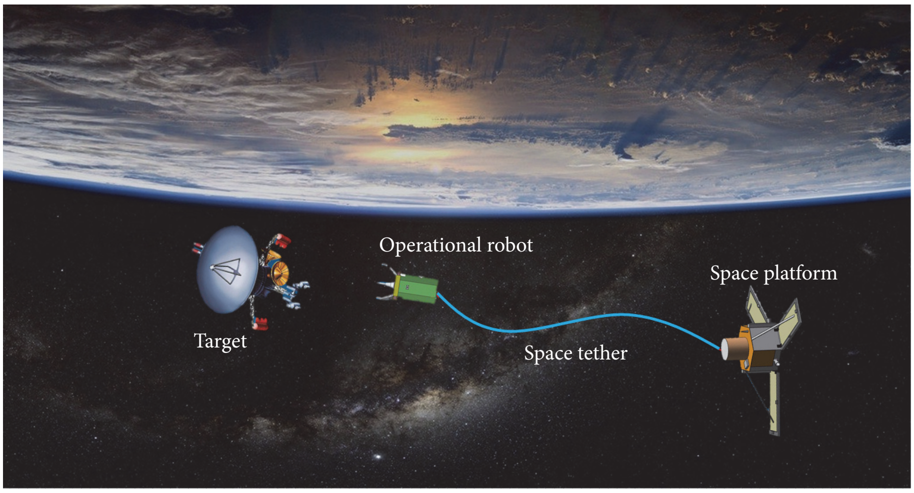


Figure 1: Diagram of tethered space robot.

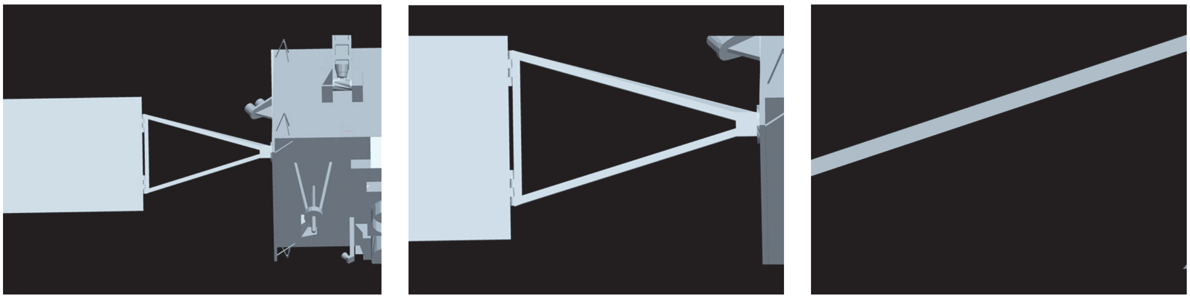
only. When it comes to noncooperative targets, any device or structure mounted on them is unknown in advance. To facilitate the determination of the grasping position, we analyze the common spacecraft structures and select satellite bracket as a suitable location for grasp, which is usually structurally robust and widely present in spacecraft [11].

During the approaching process, it is necessary for TSR to recognize the grasping position and measure their relative positions and attitudes. The calculated information will be fed into the control system to adjust TSR's self-attitude and guide its approaching path. Though microwave radar and laser radar have been used in relative navigation, we select charge-coupled device (CCD) cameras mounted on the operational robot to provide vision-based measurements considering their low mass and rich view information [12].

Providing the on-orbit service involves many key techniques and one of them is the control of the relative position and attitude between TSR and the target.

Yuga et al. [13] proposed a coordinated control method using tension and thruster. Wang et al. [14, 15] proposed that, through altering the position of the connecting point between the tether and the robot, the tether tension force was able to achieve the orbit and attitude control simultaneously. Mori and Matunaga [16] designed tether tension/rope length controller and realized the attitude control. Experiments showed that this controller could save propellants. But these methods all require that the relative position between the space platform, tethered robot, and the target is known. In order to measure the relative position and orientation between them, visual servoing control problem has been widely studied. Xu et al. [17] proposed a stereo vision-based method to measure the relative position and orientation. The basic idea is to identify the feature points of the target in 3D space. Thienel et al. [18] presented a nonlinear method to estimate the attitude of the target spacecraft and then realized the tracking to it. But it assumed that the target attitude was already estimated using vision method. Hafez et al. [19] proposed a visual servoing control method of the dual-arm space robot. In [20], Dong and Zhu used the extended Kalman filter to estimate pose and motion of noncooperative target in real time. Cai et al. [21] proposed a monocular visual servoing control system for the tethered space robot. It realized the real time tracking of noncooperative target satellite in complex scenes. For these position and orientation measurement methods, the key technology is the detection and recognition of the characteristics of the target spacecraft. Considering the wide existence of circle-shape device, Hough Transform methods [22] are generally adopted to detect space target in autonomous rendezvous and docking. Casonato and Palmerini [23] used Hough Transform to identify basic geometric shapes, such as circles and straight lines, in rendezvous monitoring. The work [24] also used Hough Transform to convert the image from spatial space to parameter space, facilitating the detection of launch bolt holes. In [25], two marks, an external circle and an internal circle, are laid on different planes of the target to indicate the relative attitude and position. They can be easily detected by Hough Transform methods too. In addition, Ramirez et al. [26] proposed a quadrilateral recognition method, which could be used to recognize the spacecraft's solar panels. Grompone Von Gioi et al. [27] proposed a line segment detection algorithm based on image gradients. It could achieve rapid detection of line segments and had been applied to detect the edge lines of the spacecraft's solar panels. Wang et al. [28, 29] proposed a new approach to dynamic eye-in-hand visual tracking for the robot manipulator in constrained environments with an adaptive controller.

The above methods generally aim at detecting the characteristics of the target from a relatively far distance, where the point features, linear features, or geometric features of the target could be fully detected [30], as shown in Figures 2(a) and 2(b). But when in extremely close distance (usually less than 1 m), these features are generally difficult to extract and are incomplete due to the limited FOV of cameras. The brackets supporting the solar panels are the element present in the majority of spacecraft, excluding the small cube satellites which often accommodate body-mounted panels. These elements can be easily recognized by cameras using



(a) Far distance (b) Close distance (c) Extremely close distance

Figure 2: Images captured in different approaching distances.

their edge lines (Figure 2(c)). But for the above-mentioned methods, simply using the detected edge line features is inadequate and they tend to fail [31].

In this paper, we propose a visual servoing control method based on the detection of margin line on the satellite brackets. Firstly, a gradient based edge line detection method is proposed to extract edges of the satellite brackets. Then, we construct the model to acquire the relative position and attitude between TSR and the target. Lastly, we integrate the PD controller to design a new visual servoing controller. It is able to control TSR's attitude to satisfy the grasping requirements. This method is appropriate for addressing the relative position and attitude problem in extremely close distance when TSR approaches the target satellite.

The remainder of this paper is organized as follows. In Section 2, we introduce the project model of edge lines. The detailed description of the line detection algorithm and servoing control method is presented in Sections 3 and 4. Section 5 illustrates and analyzes the experiments and we conclude our work in Section 6.

2. Projection Model of Edge Lines

2.1. Coordinates Definition. Observing system is composed of the operational robot and the satellite bracket. In extremely close stage, TSR moves towards the grasping position and only edge information of the brackets is available. In order to facilitate the relative measurement and future expression, the coordinate systems used in our paper are defined as follows:

(1) The coordinate system of operating robot body $O_GX_GY_GZ_G$: the origin O_G is located in the centroid of the robot; O_GX_G axis points to the direction of flight; O_GZ_G axis points to the geocenter; O_GY_G axis direction is determined by the right-hand rule.

(2) The coordinate system of the capture point $O_ZX_ZY_ZZ_Z$: the origin O_Z is located at the capture point of the target panel bracket; directions of O_ZZ_Z axis and O_CZ_C axis are same; the direction of axis O_ZX_Z is along the grasping pole of panel bracket; axis O_ZY_Z is determined by the right-hand rule.

(3) Camera coordinate system $O_CX_CY_CZ_C$: origin O_C is at the optical center of the camera, O_CZ_C axis is the optical axis, and O_CX_C axis and O_CY_C axis are horizontal axis and vertical axis which are parallel to the image plane.

(4) Image coordinate system $O_DX_DY_D$: coordinate plane of image coordinate system $X_DO_DY_D$ is on the camera's CCD imaging plane, the origin O_D is the intersection of optical axis and imaging plane, and O_DX_D axis and O_DY_D axis are parallel to the O_CX_C and the O_CY_C axis.

(5) Computer image coordinate system UV: the origin is located in the upper left corner of the digital image, U and V denote the number of columns and rows of a digital image. The relationships of aforementioned coordinate systems are illustrated in Figure 3.

Besides, we choose the widely adopted Pinhole model to represent the camera imaging relationship [32]. Set the coordinate of a space point P at camera coordinate system $O_CX_CY_CZ_C$ as (x_c, y_c, z_c); then the imaging model of the point P is represented as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dfrac{x_c}{z_c} \\ \dfrac{y_c}{z_c} \\ 1 \end{bmatrix}. \tag{1}$$

(u, v) is the coordinate of point P in the coordinate system of computer image. (u_0, v_0) is the main point of the camera. $\alpha_x = f/dx$, $\alpha_y = f/dy$, f is the focal length of the camera, and dx and dy are a pixel's physical sizes in the horizontal and vertical direction. Generally, dx is equal to dy.

If two points, $A(u_1, v_1)$ and $B(u_2, v_2)$, which locate in the computer image coordinate system UV, are considered, then the line AB can be expressed as

$$u = kv + b, \quad k = \frac{u_1 - u_2}{v_1 - v_2}, \; b = \frac{v_1u_2 - u_1v_2}{v_1 - v_2}. \tag{2}$$

2.2. Camera Imaging Model. Assume the rotation matrix and translation vector from grasping point coordinate system $O_ZX_ZY_ZZ_Z$ to camera coordinate system as R_{TC} and T,

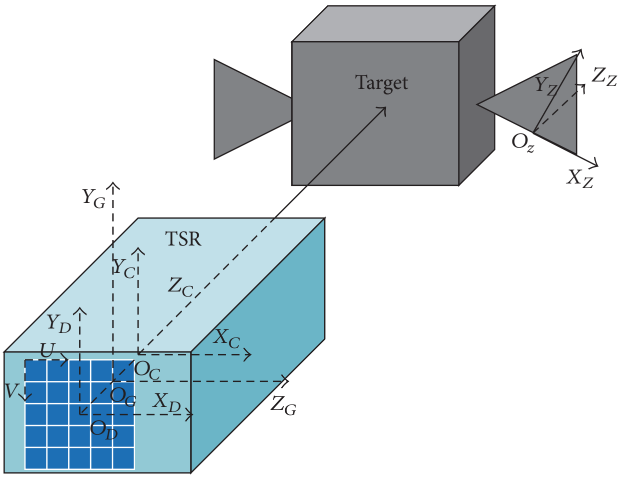


Figure 3: Definition of coordinate systems.

respectively [33]. The matrix transformed from grasping point coordinate system to camera coordinate system is

$$M_{TC} = \begin{bmatrix} R_{TC} & T \\ 0^T & 1 \end{bmatrix}. \tag{3}$$

Therefore, the transformational relationship of the space point P between homogeneous coordinates $(x_z, y_z, z_z, 1)^T$ in grasping point coordinates $O_Z X_Z Y_Z Z_Z$ and homogeneous coordinates $(x_c, y_c, z_c, 1)^T$ in the camera coordinate system $O_C X_C Y_C Z_C$ is

$$\begin{gathered}\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = M_{TC} \begin{bmatrix} x_z \\ y_z \\ z_z \\ 1 \end{bmatrix} = \begin{bmatrix} R_{TC} & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x_z \\ y_z \\ z_z \\ 1 \end{bmatrix}, \\ z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{TC} & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x_z \\ y_z \\ z_z \\ 1 \end{bmatrix}. \end{gathered} \tag{4}$$

3. Line Segment Detection

3.1. Image Gradients. Image gradients reflect the intensity changes between adjacent pixels in the image. A small gradient value corresponds to a region with smooth gray level in the image, and a large gradient value corresponds to a sharp edge region in the image. Pixels along the consistent gradient direction are likely to belong to an edge in the image. Thus, calculating the gradients of images is important for the edge structure detection. For image $\mathbf{I}$, the gradients in pixel x are defined as

$$g_x(x, y) = \frac{I(x+1, y) + I(x+1, y+1) - I(x, y) - I(x, y+1)}{2},$$

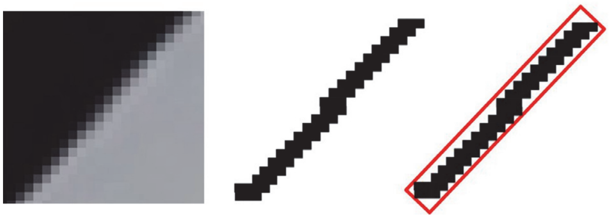


Figure 4: Line support areas.

$$g_y(x, y) = \frac{I(x, y+1) + I(x+1, y+1) - I(x, y) - I(x+1, y)}{2}, \tag{5}$$

where $I(x, y)$ represents the intensity value of pixel (x, y); $g_x(x, y)$ and $g_y(x, y)$ are horizontal and vertical gradients of pixel (x, y). Then, we calculate

$$\begin{gathered} G(x, y) = \sqrt{\left(g_x^2(x, y) + g_y^2(x, y)\right)}, \\ \theta = \arctan\left(\frac{g_y(x, y)}{g_x(x, y)}\right), \end{gathered} \tag{6}$$

where $G(x, y)$ and θ represent the gradient value and gradient orientation of pixel (x, y), respectively.

To speed up the edge structure detection and exclude the monotonous regions in image, we first preprocess the image by thresholding the gradient value of each pixel. Pixels whose gradient value is less than the predefined threshold are more likely to be from nonedge regions and will be removed directly. Beside, we define pixel consistency τ as the difference between adjacent pixel gradient orientations, which is $\tau = |\theta_1 - \theta_2|$. If $\tau < t$, it is considered that the orientations of adjacent pixels are consistent and these pixels should be in the same straight line, where t is an human-defined parameter. The determination of τ is described in Section 3.3.

3.2. Pseudo-Sorting Method. Line support area is composed of pixels with similar gradient directions in the image, as shown in Figure 4. In general, we use the external rectangle, for example, the red color one in Figure 4, to approximate the support area.

The support area of straight line segment is generated using region growing method [34, 35]. It first selects a small part of pixels as the seeds. Then, it iteratively integrates neighboring pixels with similar orientations and the seed region gradually grows. The detailed region growing strategy is introduced in Section 3.3. The performance of the region growing method is sensitive to the selection of seed pixels. One way to determine seeds is to sort image pixels according to their gradient magnitudes and selects pixels with larger values as the seed pixels. However, the computational efficiency of sorting these pixels is rather low due to the large amount of pixels contained in the image. Even though some fast sorting methods (e.g., bubble sort, quick sort) are used,

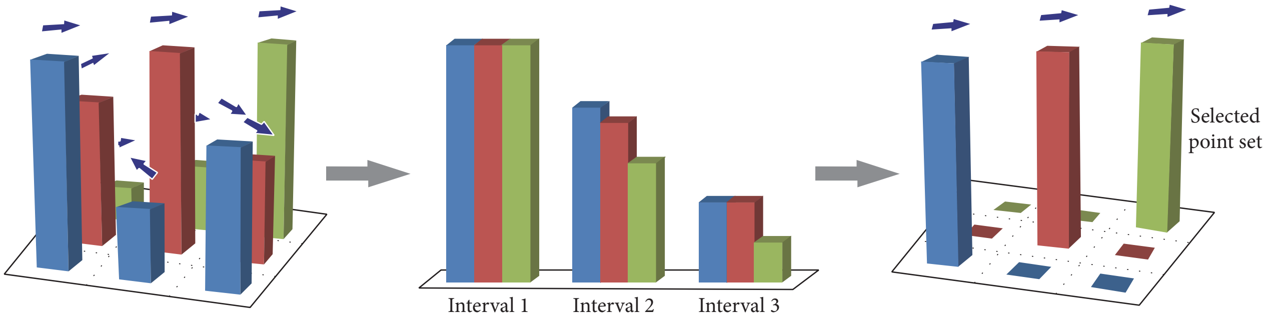


FIGURE 5: Illustration of pseudo-ordering method using a 9-pixel region.

the real time performance is still hard to obtain. Here, we adopt the pseudo-sorting method to sort the pixels (Figure 5).

Pseudo-ordering method is generally used to roughly arrange pixels in order according to their gradients. Firstly, a linear interval is generated according to the maximum and minimum values of the image gradients. Each interval represents a range of gradient values. Pixels are placed in the corresponding intervals according to their gradient values. Hence, as the increase of interval number, the pseudo-ordering results will become more accurate while the algorithmic efficiency becomes lower. Then, for pixels in the same interval, they have the same possibility of being selected as seeds.

To determine the number of intervals, we take several experiments and conclude that, for common 8-bit image, setting interval number to be 1024 could achieve a good tradeoff between performance and time complexity.

3.3. Region Growing Method. According to the results of the pseudo-sorting, the pixel point is selected as the seed point of the region growing according to the descending gradient value [36]. First, we define the direction of the line support area as follows:

$$\theta_{\text{region}} = \arctan\left(\frac{\sum_i \sin(\theta_i)}{\sum_i \cos(\theta_i)}\right), \tag{7}$$

where θ_{region} represents the angle of the support area and θ_i represents the linear direction angle of the pixel i. Set the seed point of region growing as (x, y); then the direction of the support area at the beginning is the direction of the seed point; namely, $\theta_{\text{region}} = \theta_{(x,y)}$.

Let $S_x = \cos(\theta_{\text{region}})$ and $S_y = \sin(\theta_{\text{region}})$. For each point $p(x, y)$ in the support area, we visit its 8 neighboring pixels, which are $p(x-1, y-1)$, $p(x, y-1)$, $p(x+1, y-1)$, $p(x-1, y)$, $p(x+1, y)$, $p(x-1, y+1)$, $p(x, y+1)$, and $p(x+1, y+1)$, and calculate the difference between their gradient orientations and θ_{region}. If the difference is less than τ, it is considered that they have the same direction, and the corresponding pixel will be added to the straight line support area. Then, we update the direction of the support area using

$$\theta_{\text{region}} = \arctan\left(\frac{S_y + \sin(\theta_{\overline{P}})}{S_x + \cos(\theta_{\overline{P}})}\right), \tag{8}$$

where $\overline{P}$ denotes the selected neighbor pixel of $p(x, y)$. The visited pixels will be removed from the candidate list and not be accessed again. Repeat above steps until no new pixel could be added to the linear support area, and the regional growth algorithm terminates.

During the region growing, the selection of the threshold value τ is key and will influence the final results of the algorithm. Intuitively, when a small τ is adopted, only pixels with almost identical orientations could be included in the line support area. The generated support area may be too short to provide the complete edge information. As τ increases, more nonedge pixels tend to be covered by the support area, which leads to its increase in length and width. This will cause a lot of false positive detection. In order to make a tradeoff between deficient edge information and high false positive rate, we evaluate the performance under different τ and select τ to be 22.5° by trial and error.

The region growing results under different τ are illustrated in Figure 6. Besides 22.5°, two typical thresholds, which are smaller and larger than 22.5°, are selected to provide the visual examples.

3.4. Straight Line Determination. According to the aforementioned region growing algorithm, the support area of straight line segment is generated. Then the next step is to produce the corresponding line segments from each support area. First, in order to reduce the computational burden, the support area with extremely few pixels is excluded, which is likely caused by the isolated noise. Excluding areas with few pixels can be regarded as a type of erosion operation in image processing. Both of them aim to eliminate the small regions in image. Specifically, we could use a hollow circle template with small radius and slide it over the image. For areas with few pixels, they will be totally enclosed by the template and hence will be discarded. For support region areas, they will intersect with the circle due to their length and will be retained.

In order to determine the straight line segment, the external rectangle is used to describe its support area. The support area is scored by summing up the gradient values of pixels contained in it. Then the support area centroid (c_x, c_y) can be calculated by the following formula:

$$\begin{aligned} c_x &= \frac{\sum_{j\in \text{Re}} G(j) * x(j)}{\sum_{j\in \text{Re}} G(j)}, \\ c_y &= \frac{\sum_{j\in \text{Re}} G(j) * y(j)}{\sum_{j\in \text{Re}} G(j)}, \end{aligned} \tag{9}$$

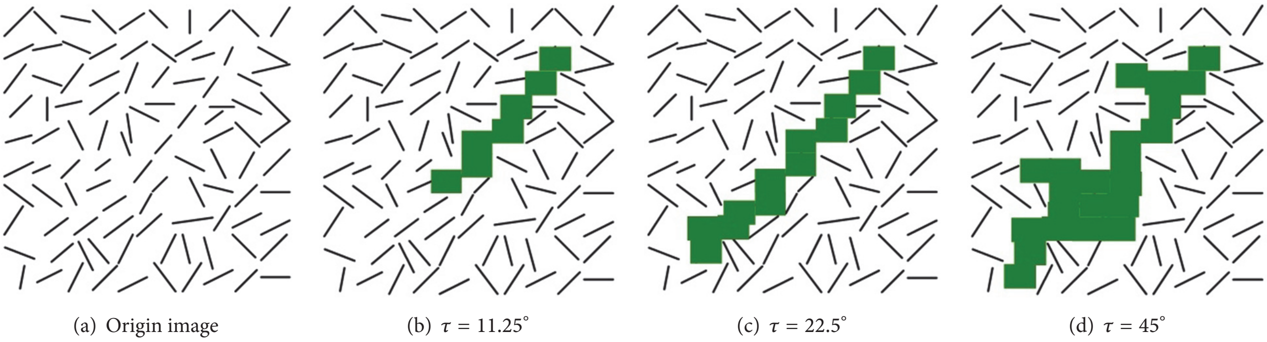

(a) Origin image (b) $\tau = 11.25^\circ$ (c) $\tau = 22.5^\circ$ (d) $\tau = 45^\circ$

Figure 6: Region growing results with different τ. (a) is the original gradient map, including the edge area with noise interference. (b), (c), and (d) are results corresponding to 11.25°, 22.5°, and 45°, respectively.

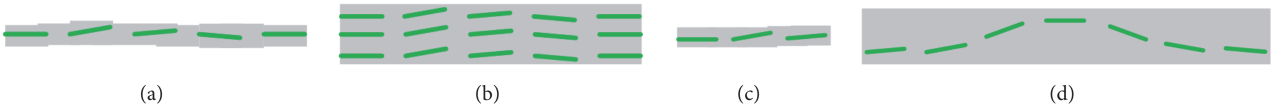

(a) (b) (c) (d)

Figure 7: Illustration of different support areas including (a) straight line, (b) monotonous region, (c) isolated noise, and (d) low-curvature curve. (b) and (c) could be discarded by criterion 1; (d) could be discarded by criterion 2.

where $G(j)$ is the gradient magnitude value of the pixel j in the support area Re; $x(j)$ and $y(j)$ are coordinates of pixel j. The center of the external rectangle is chosen as the center of the support area, and the long axis of the rectangle is chosen as the direction θ_{region} of the support area, with the short axis vertical to θ_{region}.

Since not all of the straight segment support areas correspond to a straight line segment model, they need to be further judged after obtaining the external rectangle of the support area. In this paper, we develop two criteria to achieve this.

(1) The ratio of long and short axis in the external rectangle should be larger than the set threshold (typically 4 to 7).

Due to the existence of noise, pixels in monotonous region may be wrongly selected as the seeds of region growing method, leading to the support areas shown in Figure 7(b). Besides, the noise is also likely to form some small line segments, which cannot be detected due to the pixel number it contained, as shown in Figure 7(c). It is obvious to see that by thresholding the ratio between long and short axis these support areas can both be eliminated.

(2) When the direction of the external rectangle is consistent with the direction of the long axis of rectangle (direction deviation is less than τ), the proportion of its pixel number to the total number of pixels contained in the rectangular should be larger than the predefined threshold (typically 50% to 70%).

Determining the straight lines with long-short axis ratio may include some low-curvature curves, such as the circular edge with a larger radius, as shown in Figure 7(d). It can be observed that although the curve shares a large long-short axis ratio, it tends to produce the increase of the support region area. Hence, by thresholding the proportion of the pixel number divided by the support region area, these regions can be eliminated.

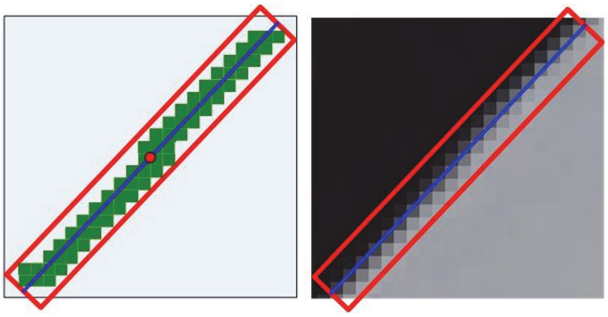


Figure 8: Schematic diagram of the line segment validation.

The support areas which satisfy these two criteria correspond to the straight line segment model, and the principal axis which comes through the center of the external rectangle is the straight line segment. As shown in Figure 8, the principal axis of the external rectangle, that is, the blue line segment, is the straight segment.

4. Servoing Controller Design

Through the above algorithm, we can detect the edge lines of the satellite brackets in the image. These operations are all based on the computer image coordinates. Hence, the detected edge lines can be expressed as

$$\begin{aligned} u &= k_1 v + b_1, \\ u &= k_2 v + b_2. \end{aligned} \tag{10}$$

According to (10), we design a controller to achieve the control of relative attitude of the robot in the ultra-close approaching stage.

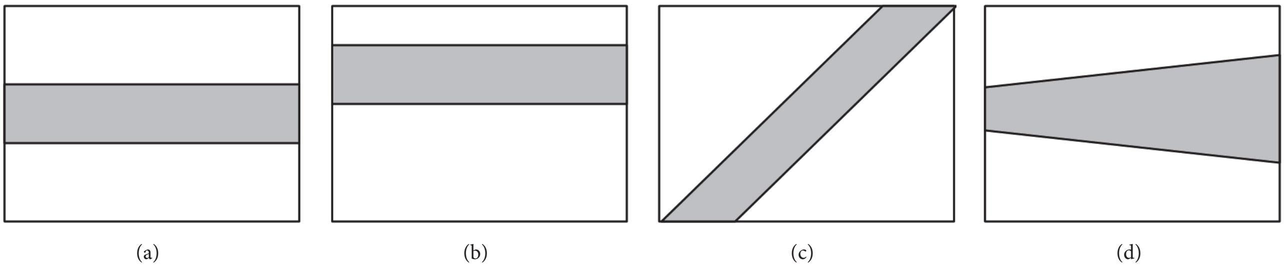


Figure 9: Relationships of two edge lines under different relative attitudes.

4.1. Edge Lines Analysis. The capturing device of TSR is usually installed in the front plate of the robot and close to the camera. Hence, we set that the capture coordinate system coincides with the camera coordinate system $O_CX_CY_CZ_C$, and assume the Euler angles between capture coordinate system $O_CX_CY_CZ_C$ and target coordinate system $O_ZX_ZY_ZZ_Z$ as α, β, γ, respectively [37]. It is easy to analyze that, for different relative attitude of the robot and the target, the imaging of the bracket edge lines in the camera image plane is shown in Figure 9.

The imaging result in Figure 9(a) shows the final state of an ideal controller. It is requested that the Euler angles β and γ should be zero, and the relative position deviation y along direction Y is zero. The Euler angles α of axis X can be any angle. Figure 9(b) is the case that y is not zero. Figure 9(c) shows that the Euler angle γ is not zero and there exists rotation along axis Z. The edge lines in Figure 9(d) are not parallel ($k_1 \neq k_2$) and it reveals that the robot rotates along axis Y.

4.2. Controller Design. From the above analysis, we can conclude that (1) the relative attitude angle γ can be measured by summing up the slopes of two edge lines; (2) the relative attitude angle β can be measured by subtracting the slopes of two edge lines; (3) the position deviation y along axis Y can be measured using the sum of b_1 and b_2. Therefore, we refer to the PD controller and design our controller as follows [38]:

$$\begin{aligned} M_z &= -p_1\left(k_1+k_2\right)-p_2\left(\dot{k}_1+\dot{k}_2\right), \\ M_y &= -p_3\left(k_1-k_2\right)-p_4\left(\dot{k}_1-\dot{k}_2\right), \\ F_y &= -p_5\left(b_1+b_2-2u_0\right)-p_6\left(\dot{b}_1+\dot{b}_2\right), \end{aligned} \tag{11}$$

where $p_1 \sim p_6$ are the controller parameters and should be adjusted in practice. M_z and M_y are the control torque of axes Z and Y. F_y is the control force of axis Y. The stability of the proposed controller could be proved by Lyapunov first method [39, 40].

We have to note that the attitude angle α is not controlled by the controller, because only the pole structure of the bracket is imaged in ultra-close distance (Figure 9(a)) and it is resistant to attitude change along axis X. But in practice α should be restricted during the approaching process using additional sensors, for example, the global camera, to keep TSR's self-attitude stable. The flowchart of our method is shown in Figure 10.

Table 1: The composition of visual perception subsystem.

Camera (VS-902H)	CCD: 1/3″ Resolution: 752 (H) × 582 (V)
Lens (Computar M1614)	Focal length: 5 mm Field of view: 65.5° (H) × 51.4° (V)
Processor	Pentium Dual-Core E5200 with 2.50 GHz, 2G RAM Operating system: Windows XP SP3 Programming environment: VS2005 IDE with OpenCV

5. Experiments and Results

5.1. Experiment Conditions. In order to verify the effectiveness of the proposed control algorithm, we set up a visual servoing control system, as shown in Figure 11. It mainly consists of visual perception subsystem, control subsystem, movement platform, and the target satellite model. The composition of visual perception subsystem is listed in Table 1. Since this paper mainly focuses on the visual servoing control in ultra-close distance, we adopt a single-pole model to substitute the satellite bracket model. In the experiment, we use the 6-DOF movement platform to simulate TSR's movement in space. The visual perception subsystem and target model are mounted on the movement platform to control their relative position and attitude.

5.2. Results Analysis. In order to evaluate the edge line detection performance in approaching process, we make a simulation and move TSR from initial position (about 3 m to the target) to grasping position (about 0.2 m to the target). Meanwhile, the single-pole model swings with an initial rate, about 10°/s. Figure 12 qualitatively demonstrates that despite the change of relative position and attitude between TSR and the target model our proposed algorithm is always able to detect the edge lines.

Figure 13 quantitatively reveals the measurement values of slope k and intercept b of the detected edge lines in every frame, respectively. In (a), the initial slope is −0.45, which means TSR's self-attitude needs to be further adjusted. As

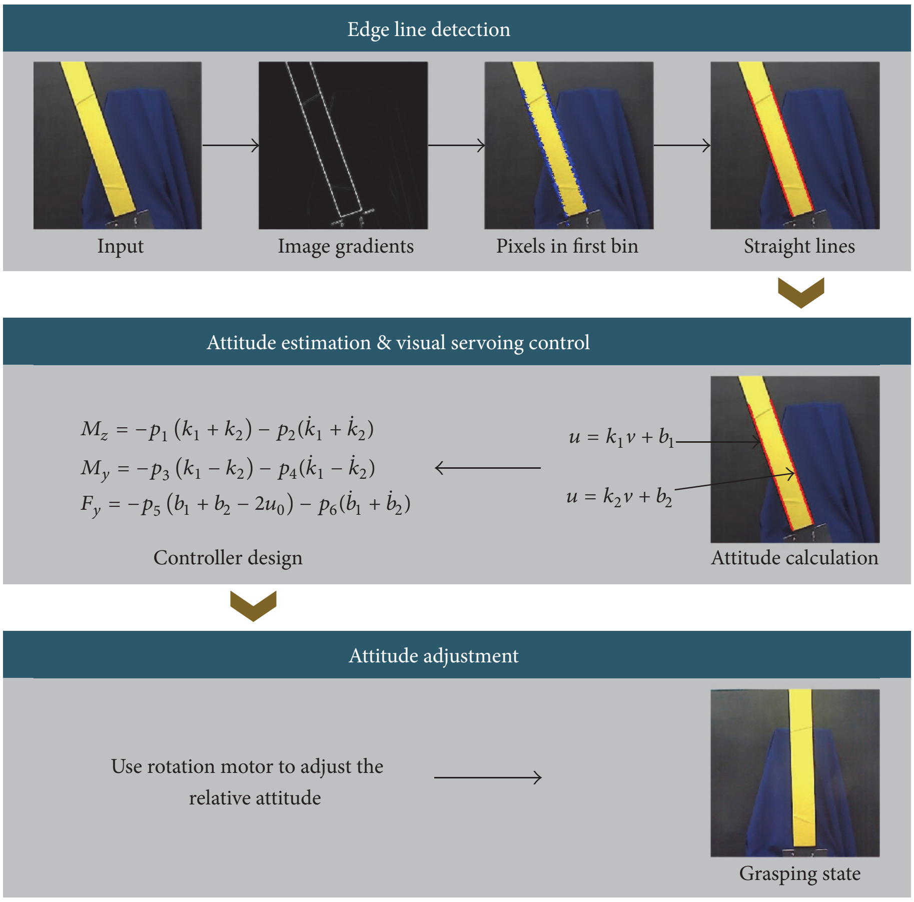
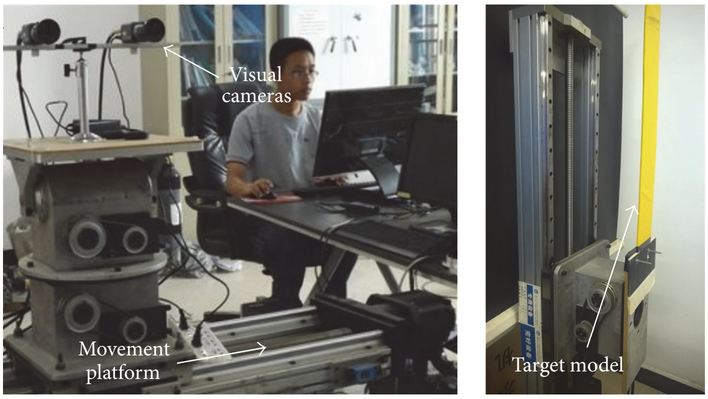

Edge line detection
Input
Image gradients
Pixels in first bin
Straight lines
Attitude estimation & visual servoing control
$M_z = -p_1(k_1 + k_2) - p_2(\dot{k}_1 + \dot{k}_2)$
$M_y = -p_3(k_1 - k_2) - p_4(\dot{k}_1 - \dot{k}_2)$
$F_y = -p_5(b_1 + b_2 - 2u_0) - p_6(\dot{b}_1 + \dot{b}_2)$
Controller design
$u = k_1 v + b_1$
$u = k_2 v + b_2$
Attitude calculation
Attitude adjustment
Use rotation motor to adjust the relative attitude
Grasping state

Figure 10: Flowchart of the proposed method.



(a) Visual perception and movement subsystems

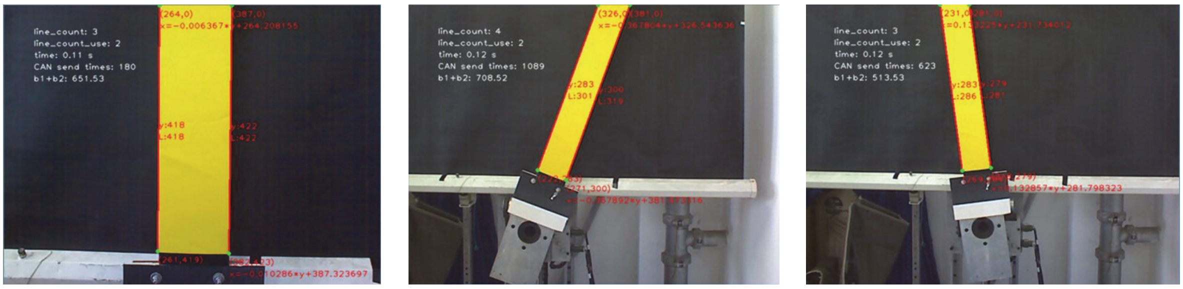
(b) Single-pole satellite model

Figure 11: Visual servoing control system.

TSR approaches the target, slops of the two detected lines gradually converge to about 0, which means TSR reaches the ideal grasping attitude. Meanwhile, intercepts of the detected lines converge to about 400 and 300, respectively. It is easy to find that their sum is rather close to the image width $2u_0$. It indicates that the two edge lines are horizontally symmetric along the image center and TSR reaches the ideal grasping position.

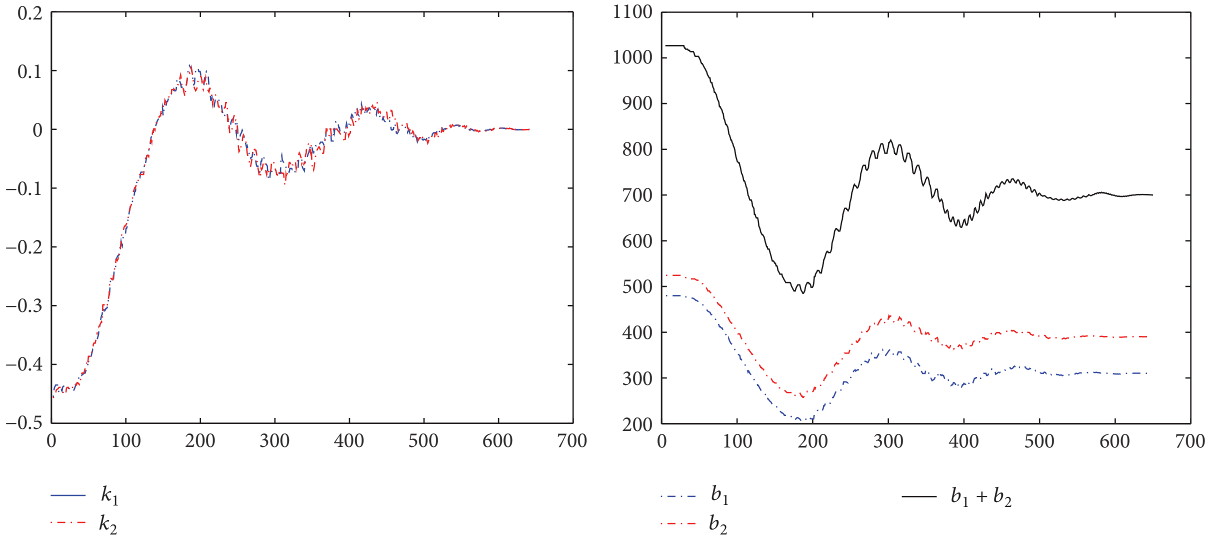
In Figure 14, we give the time consumption of the proposed method. The average time of detecting one image is about 0.115 seconds. Considering the control period of the controller is 0.25 seconds, our method could satisfy the real time need.

5.3. Limitations. As mentioned in Section 4.2, TSR's rotation around axis x is not taken into consideration in this paper.



(a) Close-range detection (b) Detection with positive slope (c) Detection with negative slope

Figure 12: Edge line detection results under the different position and attitude.



(a) (b)

Figure 13: Measurement value curves of slope k (a) and intercept b (b) of the detected edge lines. All curves converge at about 500th frame.

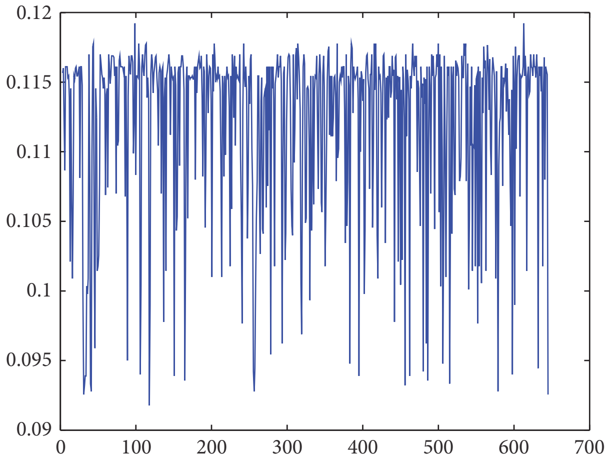


Figure 14: Comparison of time consumption per frame.

But in practice this rotation will lead to TSR's deviation from the approaching path and increase the possibility of collision with the target. Hence in future works the control to attitude angle α should be added to the controller.

Although our proposed method is able to work successfully, the number of frames it needs to offset the relative position and attitude is a bit high. It could be ascribed to the low frame frequency (4 Hz) of visual camera. The time span between adjacent frames leads to the inconsistency between actual measurement values and control parameters. In order to solve this, we consider (1) increasing the frame frequency of cameras as well as further reducing the complexity of our method and (2) introducing the motion prediction mechanism of TSR to offset the attitude change of target, in the future.

6. Conclusions

This paper presents a novel monocular visual servoing control method using only the edge lines. It is designed to measure the relative position and attitude between TSR and the target under nonfull field of view, which is intractable for traditional feature based visual servoing methods.

The novelty of this paper is constructed by the following three aspects: (1) We propose a novel edge line detection method. It is based on gradient growing and can stably extract lines in the image. Also, it meets the real time requirement of controller. (2) We build the model and analyze the relationship between edge line parameters and the relative attitude. (3) We design a PD-based visual servoing controller for ultra-close approaching process, which requires the two detected edge line parameters only.

Experiments on the semiphysical simulation system indicate that our method is invariant to rotation and scale change.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research is sponsored by the National Natural Science Foundation of China (Grant nos. 11272256, 60805034) and the Fundamental Research Funds for the Central Universities (Grant no. 3102016BJJ03).

References

[1] M. Shan, J. Guo, and E. Gill, "Review and comparison of active space debris capturing and removal methods," *Progress in Aerospace Sciences*, vol. 80, pp. 18–32, 2016.

[2] P. Huang, F. Zhang, Z. Meng, and Z. Liu, "Adaptive control for space debris removal with uncertain kinematics, dynamics and states," *Acta Astronautica*, vol. 128, pp. 416–430, 2016.

[3] F. Zhang and P. Huang, "Releasing dynamics and stability control of maneuverable tethered space net," *IEEE/ASME Transactions on Mechatronics*, vol. PP, no. 99, pp. 1–1, 2016.

[4] P. Williams, "Optimal orbital transfer with electrodynamic tether," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 2, pp. 369–372, 2005.

[5] P. Pergola, A. Ruggiero, M. Andrenucci, and L. Summerer, "Low-thrust missions for expanding foam space debris removal," in *Proceedings of the 32nd International Electric Propulsion Conference*, Wiesbaden, Germany, September 2011.

[6] P. Huang, D. Wang, Z. Meng, F. Zhang, and Z. Liu, "Impact dynamic modeling and adaptive target capturing control for tethered space robots with uncertainties," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 5, pp. 2260–2271, 2016.

[7] P. Huang, D. Wang, Z. Meng, F. Zhang, and J. Guo, "Adaptive postcapture backstepping control for tumbling tethered space robot–target combination," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 1, pp. 150–156, 2016.

[8] P. Huang, M. Wang, Z. Meng, F. Zhang, Z. Liu, and H. Chang, "Reconfigurable spacecraft attitude takeover control in post-capture of target by space manipulators," *Journal of the Franklin Institute*, vol. 353, no. 9, pp. 1985–2008, 2016.

[9] P. Huang, D. Wang, F. Zhang, Z. Meng, and Z. Liu, "Postcapture robust nonlinear control for tethered space robot with constraints on actuator and velocity of space tether," *International Journal of Robust and Nonlinear Control*, 2016.

[10] L. Chen, P. Huang, J. Cai, Z. Meng, and Z. Liu, "A non-cooperative target grasping position prediction model for tethered space robot," *Aerospace Science and Technology*, vol. 58, pp. 571–581, 2016.

[11] B. Bischof, L. Kerstein, J. Starke, H. Guenther, and W. Foth, "ROGER—robotic geostationary orbit restorer," *Science and Technology Series*, vol. 109, pp. 183–193, 2004.

[12] R. B. Friend, R. T. Howard, and P. Motaghedi, "Orbital express program summary and mission overview," in *SPIE Defense and Security Symposium*, Orlando, Fla, USA, April 2008.

[13] N. Yuga, S. Fumiki, and N. Shinichi, "Guidance and control of 'tethered retriever' with collaborative tension-thruster control for future on-orbit service missions," in *Proceedings of the 8th International Symposium on Artificial Intelligence: Robotics and Automation in Space (i-SAIRAS '05)*, Munich, Germany, September 2005.

[14] D. Wang, P. Huang, J. Cai, and Z. Meng, "Coordinated control of tethered space robot using mobile tether attachment point in approaching phase," *Advances in Space Research*, vol. 54, no. 6, pp. 1077–1091, 2014.

[15] D. Wang, P. Huang, and Z. Meng, "Coordinated stabilization of tumbling targets using tethered space manipulators," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 2420–2432, 2015.

[16] O. Mori and S. Matunaga, "Formation and attitude control for rotational tethered satellite clusters," *Journal of Spacecraft and Rockets*, vol. 44, no. 1, pp. 211–220, 2007.

[17] W. Xu, B. Liang, C. Li, and Y. Xu, "Autonomous rendezvous and robotic capturing of non-cooperative target in space," *Robotica*, vol. 28, no. 5, pp. 705–718, 2010.

[18] J. K. Thienel, J. M. VanEepoel, and R. M. Sanner, "Accurate state estimation and tracking of a non-cooperative target vehicle," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 5511–5522, Keystone, Colo, USA, August 2006.

[19] A. H. A. Hafez, V. V. Anurag, S. V. Shah, K. M. Krishna, and C. V. Jawahar, "Reactionless visual servoing of a dual-arm space robot," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '14)*, pp. 4475–4480, June 2014.

[20] G. Dong and Z. H. Zhu, "Position-based visual servo control of autonomous robotic manipulators," *Acta Astronautica*, vol. 115, pp. 291–302, 2015.

[21] J. Cai, P. Huang, L. Chen, and B. Zhang, "An efficient circle detector not relying on edge detection," *Advanced in Space Research*, vol. 51, no. 11, pp. 2359–2375, 2014.

[22] H. Yuen, J. Princen, J. Illingworth, and J. Kittler, "Comparative study of hough transform methods for circle finding," *Image and Vision Computing*, vol. 8, no. 1, pp. 71–77, 1990.

[23] G. Casonato and G. B. Palmerini, "Visual techniques applied to the ATV/ISS rendez-vous monitoring," in *Proceedings of the IEEE Aerospace Conference Proceedings*, pp. 613–624, Big Sky, Mont, USA, March 2004.

[24] M. Aull, "Visual servoing for an autonomous rendezvous and capture system," *Intelligent Service Robotics*, vol. 2, no. 3, pp. 131–137, 2009.

[25] M. Sabatini, G. B. Palmerini, and P. Gasbarri, "A testbed for visual based navigation and control during space rendezvous operations," *Acta Astronautica*, vol. 117, pp. 184–196, 2015.

[26] V. A. Ramirez, S. A. M. Gutierrez, and R. E. S. Yanez, "Quadrilateral detection using genetic algorithms," *Computación y Sistemas*, vol. 15, no. 2, pp. 181–193, 2011.

[27] R. Grompone Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: a fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2010.

[28] H. Wang, Y.-H. Liu, W. Chen, and Z. Wang, "A new approach to dynamic eye-in-hand visual tracking using nonlinear observers," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 2, pp. 387–394, 2011.

[29] H. Wang, B. Yang, Y. Liu, W. Chen, X. Liang, and R. Pfeifer, “Visual servoing of soft robot manipulator in constrained environments with an adaptive controller,” *IEEE/ASME Transactions on Mechatronics*, 2016.

[30] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, “A review of space robotics technologies for on-orbit servicing,” *Progress in Aerospace Sciences*, vol. 68, pp. 1–26, 2014.

[31] J. Liu, N. Cui, F. Shen, and S. Rong, “Dynamics of RObotic GEostationary orbit restorer system during deorbiting,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 29, no. 11, pp. 36–42, 2014.

[32] A. Petit, E. Marchand, and K. Kanani, “Vision-based space autonomous rendezvous: a case study,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’11)*, pp. 619–624, IEEE, San Francisco, Calif, USA, September 2011.

[33] B. P. Larouche and Z. H. Zhu, “Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control,” *Autonomous Robots*, vol. 37, no. 2, pp. 157–167, 2014.

[34] J.-P. Gambotto, “A new approach to combining region growing and edge detection,” *Pattern Recognition Letters*, vol. 14, no. 11, pp. 869–875, 1993.

[35] S. A. Hojjatoleslami and J. Kittler, “Region growing: a new approach,” *IEEE Transactions on Image Processing*, vol. 7, no. 7, pp. 1079–1084, 1998.

[36] K.-L. Chung, Y.-H. Huang, J.-P. Wang, T.-C. Chang, and H.-Y. Mark Liao, “Fast randomized algorithm for center-detection,” *Pattern Recognition*, vol. 43, no. 8, pp. 2659–2665, 2010.

[37] E. Rosten, R. Porter, and T. Drummond, “Faster and better: a machine learning approach to corner detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010.

[38] M. Nohmi, “Mission design of a tethered robot satellite “STARS” for orbital experiment,” in *Proceedings of the IEEE International Conference on Control Applications (CCA ’09)*, pp. 1075–1080, St. Petersburg, Russia, July 2009.

[39] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck, “Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle,” *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 743–749, 2009.

[40] E. Malis, F. Chaumette, and S. Boudet, “2 1/2 D visual servoing,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.

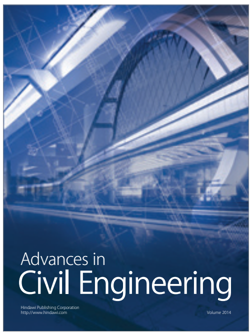
Journal of
Engineering

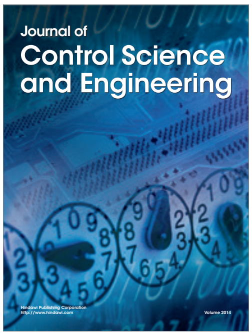
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

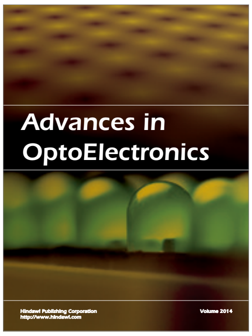
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

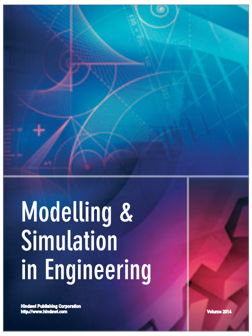
Journal of
Robotics

Journal of
Electrical and Computer Engineering

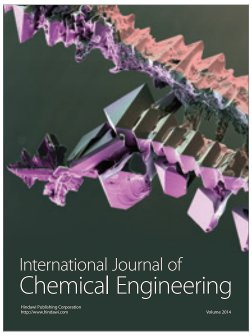

Advances in
OptoElectronics

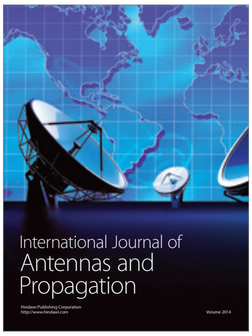
VLSI Design

International Journal of
Navigation and Observation

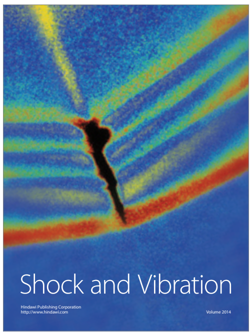

Modelling & Simulation in Engineering

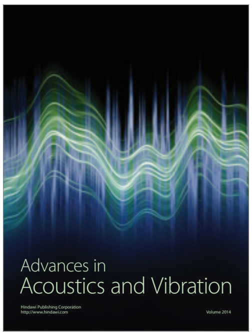
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration