sensors

MDPI

Article

A TSR Visual Servoing System Based on a Novel Dynamic Template Matching Method [†]

Jia Cai [1,2], Panfeng Huang [1,2,*], Bin Zhang [1,2] and Dongke Wang [1,2]

Received: 28 October 2015; Accepted: 10 December 2015; Published: 21 December 2015
Academic Editors: Lianqing Liu, Ning Xi, Wen Jung Li, Xin Zhao and Yajing Shen

[1] National Key Laboratory of Aerospace Flight Dynamics, Northwestern Polytechnical University, 710072 Xi'an, China; caijia@mail.nwpu.edu.cn (J.C.); zhangbindt@mail. nwpu.edu.cn (B.Z.); wdkly.student@gmail.com (D.W.)

[2] Research Center for Intelligent Robotics, School of Astronautics, Northwestern Polytechnical University, 710072 Xi'an, China

[*] Correspondence: pfhuang@nwpu.edu.cn; Tel.: +86-29-88460366 (ext. 801)

[†] This paper is an extended version of the paper entitled "Novel Dynamic Template Matching of Visual Servoing for Tethered Space Robot" presented at ICIST, 2014, Shenzhen, China, 26–28 April 2014.

Abstract: The so-called Tethered Space Robot (TSR) is a novel active space debris removal system. To solve its problem of non-cooperative target recognition during short-distance rendezvous events, this paper presents a framework for a real-time visual servoing system using non-calibrated monocular-CMOS (Complementary Metal Oxide Semiconductor). When a small template is used for matching with a large scene, it always leads to mismatches, so a novel template matching algorithm to solve the problem is presented. Firstly, the novel matching algorithm uses a hollow annulus structure according to a FAST (Features from Accelerated Segment) algorithm and makes the method be rotation-invariant. Furthermore, the accumulative deviation can be decreased by the hollow structure. The matching function is composed of grey and gradient differences between template and object image, which help it reduce the effects of illumination and noises. Then, a dynamic template update strategy is designed to avoid tracking failures brought about by wrong matching or occlusion. Finally, the system synthesizes the least square integrated predictor, realizing tracking online in complex circumstances. The results of ground experiments show that the proposed algorithm can decrease the need for sophisticated computation and improves matching accuracy.

Keywords: Tethered Space Robot; object recognition; template matching; visual servoing; target tracking

1. Introducation

1.1. Motivation

Nowadays, on-orbit service technologies are drawing more and more attention, since they have many potential values for novel applications, such as space debris cleaning, on-orbit assembly, and maintenance, as shown in [1–4]. In the last few decades, almost all robotics used in space have the same mechanical elements. For on-orbit service tasks, they are usually modelled as a manipulator mounted on a rigid body structure. Devices such as the Orbital Express, the Shuttle Remote Manipulator System (Canadarm), and the Japanese ETS-VII are examples. However, limited by the manipulator length their flexibility seems to not be enough for some novel emerging space tasks [5,6].

Aiming to overcome these rigid manipulator arms' shortcomings, our proposed Tethered Space Robot (TSR) system is designed to be different from traditional robotics for space manipulation, as shown in [7]. As a novel type of space robot, it has a creative architecture, whereby the TSR comprises an Operational Robot, a 100 meter-long Space Tether and a Robot Platform, as shown in Figure 1. TSR has a large operational radius that benefits from the 100 m tether, which also enhances its flexibility, making it an especially promising technology for non-cooperative object capture.

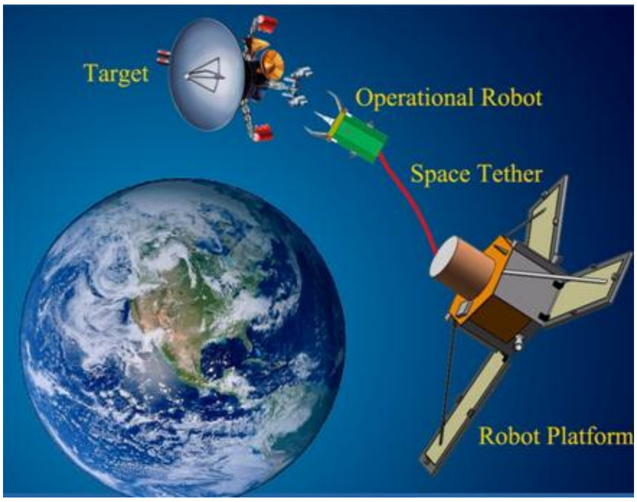


Figure 1. Description of the tethered space robot system.

In autonomous robot tracking and capture of a non-cooperative target, computer vision is exclusively used as the primary feedback sensor in these tasks to control the pose of the end-effector with respect to the target, as described in [8–10]. This paper concentrates on calculating the target's azimuth angles based on tracking the region of interest (ROI), which is selected for capture manipulation. According to [11,12], there are common parts mounted on non-cooperative targets, such as the apogee rocket motor injector, the docking ring, the separator bolts, the span of solar panels and so on. Since they are in common use, we also select them as the target for the Operational Robot's capture manipulation and we choose the span of solar panels as our research interest in this paper.

1.2. Challenge Descriptions

Many hypothetic methods from the literature can be applied to optimize the association process. Target observations can be acquired using recognition and tracking methods, such as template-based or feature-based ones. Unfortunately, the span of solar panels is mostly made by metal with a smooth surface in our scenario. Sometimes they are even covered with a burnished membrane for thermal control. As described in great detail in [13], it is difficult to extract and track feature points from such a smooth region, and even using FAST, ORB (Oriented FAST and Rotated BRIEF) can only detect few corners. Furthermore, the extracted features are confused and disordered, and cannot be selected as ROI for capture manipulation. In addition, tracking algorithms such as Meanshift, CAMSHIFT, based on color histograms will not work well since there are insufficient colors and the solar panels are very similar to the satellite's main body.

In this paper, we focus on template matching techniques. The template matching approach is derived from the idea of searching for assigned image sections or given features within an interrelated image [14]. The template can occupy only a limited image area, or it can have the same size as the search image. The matching function is accomplished by exploiting a correlation relationship. Researchers have proposed many kinds of correlation laws, among which Sum of Hamming Distances (SHD), Normalized Cross Correlation (NCC), Sum of Absolute Difference (SAD), Sum of Squared Difference (SSD), and distance transformation ones are the most commonly used.

In view of the above analysis, a template matching algorithm is worth considering to deal with this troublesome problem. In general, standard template matching has four advantages: simplicity and accuracy, high applicability and good anti-noise performance.

When computing azimuth angles for a far-distance approach, one crucial task is how to recognize the target's ROI and track it accurately and stably. Once given a small image of the non-cooperative object, as small as 35 pixels × 35 pixels, the problem becomes more difficult. A detailed description of this problem is that the algorithm is requested to recognize a preset ROI online in dynamic scenes precisely and robustly. Between consequent frames, there are several challenges, such as motion blur, noise level, target's whirl movement and so on. For our target, which has a monotonous texture and simple structure, this becomes even more troublesome.

1.3. Related Works

Given that ROI extraction is outside the scope of this paper, we assume that the initial template object to be recognized and tracked can be extracted by some popular recognition algorithm or selected manually.

Recently, video cameras and related methods for image processing have become a standard perception system and technique for robotics [15]. Based on visual perception and feedback, the robot control is commonly known as visual servoing. Visual servoing deals with the control problem and robot motion using visual data [16]. It is a complex problem since there many techniques are involved such as kinematics and dynamics, the control theory, image processing, and real-time computational operations [17]. According to [18], visual servoing systems can be divided into three categories:

(a) Position-based visual servoing (PBVS). Such a system is required to retrieve three-dimensional (3D) information about the scene. Researchers usually use a known camera model to measure relative six-dimensional (6D) information of the target, including its 3D position by coordinate transformation and 3D velocity by filters;

(b) Image-based visual servoing (IBVS). Such a system is required to calculate the displacement of the robot in axis x and y based on a two-dimensional (2D) image measurement technique;

(c) "2½D" visual servoing. Such a system is a method that combines the above two categories.

PBVS can acquire accurate positioning information, but at high computational cost, while IBVS is faster, but may lead to undesired trajectories. In this paper, 3D information about the scene are not required, but 2D measurements in the image are necessary. A monocular camera is used to acquire the target's ROI for azimuth computation. The camera does not need to be calibrated, so the IBVS scheme based on template matching is worth considering for solving the problems before our eyes, similar to the work of Herzog *et al.* [19].

In the past the template matching technique has been widely used for object recognition and tracking problems. For example, aiming to solve special technical problems in different application scenarios, Li *et al.* [20], and Tiirkan *et al.* [21] presented different template matching improvements. However, in our view, it should be able to deal with drastic illumination changes and shape deformations. Moreover, the size of template image must be selected as 30%–50% of the scene image. Although many researchers have addressed it for many years, the task is far from trivial. You *et al.* [22] improved the rotation invariance of the template matching algorithm by using a mean absolute difference method. Since assigned regions in reference and query images are arranged into a circular structure, the accumulated error may be brought out when confronting complex environments. Furthermore, the authors did not use a prediction filter to decrease the computational complexity and improve the accuracy. Lu *et al.* [23] studied dynamic template matching. Their method was designed for recognition and tracking air moving targets. In that paper, the authors used standard template matching, which is not a rotation-invariant method, and their experimental results showed some mismatching. Paravati *et al.* [24] improved the performance of the template matching-based tracking method. They used a relevance-based technique to reduce the domain space complexity. Furthermore, they studied a tracking filter and they used a dynamic threshold to reduce the number of points to be processed. Then they carried out some tests on image sequences from different public datasets. Their results showed that the proposed scheme could reduce the time consumption of the reference method while it can ensure the robustness. Roberto *et al.* [25] presented a 3D template matching method used for autonomous pose measurement. They restrained the pose search space to a three degree of freedom (3-DOF) database concerning only the attitude parameters. The templates are built on-line before correlating them with the acquired dataset. This method can reduce the time consumption and cut down on data storage, but the computational cost is still a big problem while ensuring constant results at the same time.

In order to circumvent the above limitations of existing methods, we propose a novel dynamic template matching strategy. Firstly, we define a similarity function named Normalized SAD. Then we improve the matching criterion by constructing a series of hollow annulus structures from square regions and counting both gray and gradient values. Thirdly, a least square integrated predictor is adopted in each frame. Simultaneously, we design a fast matching process and an updating of dynamic template strategy to ensure robustness. Finally, we use it for a TSR's monocular visual servoing scheme and perform the necessary ground tests to verify the method's performance and efficiency.

A preliminary version of this work was reported in [26]. The current work is an extended version. The remainder of this paper is organized as follows: Section 2 describes the novel dynamic template matching algorithm in detail. Section 3 designs a visual servoing controller. In Section 4, we describe experiments performed on ground and analyze the results. Concluding remarks are drawn in Section 5.

2. Methodology

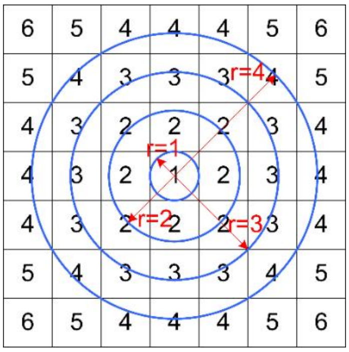
2.1. Improved Template Matching

We have introduced some common correlation-based similarity functions in the section above, namely SSD, NCC, SAD and SHD. In contrast with others, SAD is less complex and more accurate. However, its performance is limited when reference and query images have almost the same grey distribution. To improve its resolution, we define a similarity function called Normalized SAD (NSAD), shown in Equation (1):

$$d(A,B) = \sum_{x}\sum_{y} \frac{|(A(x,y) - A_m) - (B(x,y) - B_m)|}{\max(A(x,y), B(x,y))} \tag{1}$$

where, $A(x, y)$, $B(x, y)$ are the gray value of position (x, y) in the reference and query images. A_m, B_m are the mean values of the same region in the reference and query images.

After a searching process, when the value $d(A, B)$ is minimum, the best matching position is acquired. It can be found that NSAD has translation invariance by analyzing the matching criterion function. However, it lacks rotation invariance, and it may bring about mismatching when confronting complex circumstances.



6	5	4	4	4	5	6
5	4	3	3	3	4	5
4	3	2	2	2	3	4
4	3	2	1	2	3	4
4	3	2	2	2	3	4
5	4	3	3	3	4	5
6	5	4	4	4	5	6

Figure 2. Schematic diagram of distributed pixels assembled into a hollow annulus.

Considering the FAST [27] algorithm adopts a hollow annulus for detection corners and obtains a good result, we similarly create a series of concentric rings from a square region for reference, as shown in Figure 2. Given a circular structure has the quality of central symmetry, hence we propose a ring-shape criterion that can ensure the matching method is rotation-invariant. It is worth mentioning that the matching criterion based on a solid circular structure has the same performance, however, it is apt to generate accumulated errors when confronting a varied environment. Furthermore, it may lose some detailed information caused by the sum of difference values. For these reasons, some mismatching results may be obtained, as seen in [28].

Given the above factors, we construct a series of concentric complete and incomplete rings, whose radius is $r = 1, 2, \ldots, 18$. A 35 pixels × 35 pixels square region can be regarded as composed of these concentric rings. Similarly, we can deal with an N pixels × N pixels square region in the same way. According to this idea, we define a matching criterion based on gray value. For e_{oi}, i means the i-th sub concentric ring and it is the sum of the grey difference values on this ring:

$$e_{oi} = \frac{\left|\sum\limits_{j=1}^{N_i} (A_j(x,y) - A_{im}) - \sum\limits_{j=1}^{N_i} (B_j(x,y) - B_{im})\right|}{\max(A_j(x,y), B_j(x,y))} \tag{2}$$

Then we define the whole matching function by Equation (3):

$$M_o = \sum_{i=1}^{18} e_{oi} \tag{3}$$

where $A_j(x, y)$ and $B_j(x, y)$ are the grey value of the j-th coordinate on the i-th ring of reference and query images. A_{im} and B_{im} represent the mean grey value of the i-th ring. N_i is the total number of all the pixels contained on the i-th ring, while M_o is defined as the matching function of 35 pixels × 35 pixels square image based on grey value.

From the results obtained with Equation (3), we find that it also loses some information about the image's details caused by accumulated difference values. Hence, it is necessary to add some more features to describe the region more accurately and robustly. This draws our attention to the fact that the gradient feature is usually used for extracting edges. The explanation is that it represents the grey change between adjacent pixels. Hence, we try to combine gradient information with gray value for a detailed description in the matching function benefitting from the increasing pixel's uniqueness.

For pixel (x, y) of image I, we can calculate its gradient $g(x, y)$ by Equation (4):

$$g(x,y) = [I(x+1,y) - I(x-1,y)] * i + [I(x,y+1) - I(x,y-1)] * j \tag{4}$$

where, $I(x, y)$ represents the grey value of pixel (x, y), i, j is its coordinate. Meanwhile, the gradient module $m_g(x, y)$ can be calculated by Equation (5):

$$m_g(x,y) = \sqrt{g(x,y,1)^2 + g(x,y,2)^2} \tag{5}$$

where, $g(x, y, 1)$ and $g(x, y, 2)$ represent the gradient value of pixel (x, y) in direction i and j. Then we define a function using the gradient module with Equation (6):

$$e_{gi} = \sum_{j=1}^{N_i} \frac{\left|(m_{jga}(x,y) - m_{igam}) - (m_{jgb}(x,y) - m_{igbm})\right|}{\max(m_{jga}(x,y), (m_{jgb}(x,y))} \tag{6}$$

$$M_g = \sum_{i=1}^{18} e_{gi} \tag{7}$$

where, e_{gi} is the sum of the gradient difference values of the i-th sub concentric ring, $m_{jga}(x, y)$ and $m_{jgb}(x, y)$ represent the gradient modules of the j-th coordinate in the i-th ring of reference and query images. m_{igam} and m_{igbm} represent the mean gradient module of the i-th ring. M_g represents the matching function of 35 pixels × 35 pixels square region based on gradient module.

Thus, M_o and M_g are combined together and we construct a matching criterion M_h named novel NSAD function:

$$M_h = W_1 \times M_o + W_2 \times M_g \tag{8}$$

where W_1 and W_2 are the weight coefficients and their values are decided according to the different circumstances. From the above analysis, it can be concluded that our synthesized criterion M_h is rotation-invariant like criteria M_o and M_g. Meanwhile, matching accuracy can be improved since it contains more details. Hence, it can provide support to recognize targets even when confronting an inconstant environment.

2.2. Coarse to Fine Matching Strategy

The process of matching using template images can be described as a solid window sliding in a flat surface to find a similar region. For image *I*, its size is defined as $W \times H$. The size of the square template image is defined as $M \times M$. The searching step increment is usually set as one by one, hence, the sliding count is $(W - M) \times (H - M)$. Assuming *W* = 640 pixels, *H* = 480 pixels and *M* = 50 pixels, we can calculate that the original count is 253,700. To decrease the computational complexity, we design a two-steps (coarse to fine) matching strategy. Firstly, the searching step is set at *N* by *N* (*N* = 5–50). A coarse matching result can thus be acquired, and the matching position (x_c, y_c) is recorded. Secondly, we define a square region, whose center is (x_c, y_c) and side length is *N*. Let the template image slide in the square region one step at a time. When the similarity is smallest, the matching position is best. Through the coarse to fine matching strategy, we can reduce calculation counts and save time immediately. For example, given *N* = 20, the improved counts becomes 1168 (768 + 400), which is only about 4.6% of the original counts.

2.3. Least Square Integrated Predictor

When our target to be tracked is a moving one in the scene, it is essential to predict its position in the next frame using the past information since the use of predicted positions can improve occlusion invariance and decrease time consumption. Some algorithms, such as the Kalman filter, particle filter, *etc.*, have been developed to predict variance. In general, for these filter algorithms, the higher the accuracy is, the more inefficient it is. In this paper, the method is expected to perform in an online application, so we adopt a least squares integrated predictor to acquire the position in the next frame.

We define ($x(t)$, $y(t)$) as the central coordinate of the target to be tracked. The relative movement between the target and our Operational Robot is assumed to be a uniform straight-line motion. Then we can represent the best linear estimation of $x(t)$ using the following equation:

$$X_l(t) = a_0 + a_1 t \tag{9}$$

which can be rewritten as $X_l = \begin{bmatrix} 1 & t \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$,

Define $X_l(t_i)(i = 1, 2, \cdots, N)$ as the measured value in different moments. Then we can acquire the difference between the estimated value and the measured information value by Equation (10):

$$\varepsilon_i = (a_0 + a_1 t_i) - X_l(t_i) \tag{10}$$

We then calculate its *N* moments residual sum of squares:

$$\sum_{i=1}^{N} \varepsilon_i^2 = \sum_{i=1}^{N} \left[(a_0 + a_1 t_i) - X_l(t_i) \right]^2 \tag{11}$$

Through the least squares method, we can obtain the best approximation. The solution is acquired only when Equation (12) achieves the minimum value:

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \dfrac{\sum\limits_{i=1}^{N} t_i^2 \sum\limits_{i=1}^{N} X_l(t_i) - \sum\limits_{i=1}^{N} t_i \sum\limits_{i=1}^{N} X_l(t_i) t_i}{D} \\ \dfrac{-\sum\limits_{i=1}^{N} t_i \sum\limits_{i=1}^{N} X_l(t_i) + N \sum\limits_{i=1}^{N} X_l(t_i) t_i}{D} \end{bmatrix} \tag{12}$$

$$D = N\sum_{i=1}^{N} t_i^2 - \left(\sum_{i=1}^{N} t\right)^2 \tag{13}$$

Given the condition of least error, Equation (13) shows a general solution of the N moments optimum linear prediction. Similarly, if the relative movement between the target and our Operational Robot is assumed as a uniform acceleration curvilinear motion, we can give the best square estimation of $x(t)$ using Equation (14):

$$Xq = a_0 + a_1 t + a_2 t^2 \tag{14}$$

Equally, the general solution of a_0, a_1, a_2 can also be calculated by the least squares method. To consider this more deeply, we can find that the relative motion can be always divided into linear motion and curvilinear motion together in an actual scenario. On the one hand, a linear approximation can reflect the target's state quickly. On the other hand, a square approximation can smooth the target's trajectory. Hence, we consider constructing a least squares integrated predictor using them together. The function $X(t)$ is shown in Equation (15):

$$X(t) = W_4 \times X_l(t + 1/t) + (1 - W_4) \times X_q(t + 1/t) \tag{15}$$

Selection of weight coefficient W_4 for the linear and square predictor can be decided by the prediction errors. Our proposed integrated predictor possesses the property of automatic corrective measures. Furthermore, it can fit non-uniform growing curves. Through many tests, we find that W_4 = 0.8 makes the results fine in our scenario. Tests also indicate that its prediction accuracy declines when N grows under a threshold. When we choose $X(k+1) = (4x(k) + x(k-1) - 2x(k-2))/3$ (a three moments linear predictor) combined with $X(k+1) = (9x(k) - 4x(k-2) - 3x(k-3) + 3x(k-4))/5$ (a five moments square predictor), the best result is acquired.

Once the estimation coordinate is acquired, a rectangular candidate window R_c is defined for searching the template image. The width and the height are both twice those of the template image. Hence, it does not need to search the template in the whole image, which decreases the computation complexity and the time consumption.

2.4. Dynamic Template Updating Strategy

2.4.1. Updating Strategy

When approaching from 100 m to 20 m, due to the pose variation in a complex and sometimes changing environment, a non-cooperative satellite's shape and size change quickly. This phenomenon gradually makes template images very different from the target images. After several frames, the difference will be accumulated to a threshold value, and the template image used from the start may become invalid. To overcome the accumulated deviation and ensure stable matching, the template image has to be updated dynamically in time.

The matched position is defined as the geometrical center of a rectangular matching region in our method. First of all, we define two updating criteria according to the deviation of distance and similarity:

(a) The coordinate distance between a target's predicted position (x', y') and practically matched position (x, y) is defined as DIS $= \sqrt{(x - x')^2 + (y - y')^2}$.

(b) The similarity function between reference and query images is defined as $S = 1/M_h$.

Based on the above two definitions, we then design an adaptive dynamic template updating strategy. The following criteria are judged in each frame:

(a) Once *DIS* > $\mathrm{D_t}$, it means there may be an occlusion or sudden change. The position tracking does not work well any more. In order to keep tracking stably, our strategy is to use the template image continually. Meanwhile, the predicted position (x', y') is temporarily adopted as the matched coordinate (x, y) until the tracking sequence becomes fine.

(b) Once *DIS* < $\mathrm{D_t}$ and S < $\mathrm{S_t}$, it means matched position is close to the actual location. However, the difference of similarity between reference and query images is a little bigger and the tracking target is apt to be lost if we continue to use the assigned template image. Then we consider changing the template image from this frame using Equation (16):

$$TM_t = W_3 \times TM_{t-1} + (1 - W_3)P_t \tag{16}$$

where, TM_{t-1} represents the template image used in the prior frame while TM_t represents the one in the current frame. P_t means the matched region in the prior frame. Through this formula, we can update the template image adapting to changes. It is worth mentioning that $\mathrm{D_t}$, $\mathrm{S_t}$ is decided by the object's appearance and environment, *etc.*

2.4.2. Anti-Occlusion Strategy

When the object is seriously occluded, mismatching results happen, and the observation cannot be used for updating by the least squares integrated predictor. Once the object appears in the scene again, the method is expected to track it continually. To deal with this problem, we have designed an anti-occlusion strategy.

The similarity function computed in these frames is much larger than those computed in normal tracking frames. The difference d_M can be used for a passive updating strategy. Once the difference d_M > T_d, the searched sub-image is viewed as a mismatched one. Then the search range changes from the rectangular candidate window R_c to the whole image. Once d_M < T_d, the least squares integrated predictor works again and then the method narrows down the search range to the rectangular candidate window R_c again.

2.5. Theory of Calculating Azimuth Angles

In order to analyze the results more easily, we draw the pinhole camera model shown in Figure 3, where *oxy* is the Retinal Coordinate System (RCS), *o′uv* is the Pixel Coordinate System (PCS), *OcXcYcZc* is the Camera Coordinate System (CCS), and *OwXwYwZw* is the World Coordinate System (WCS).

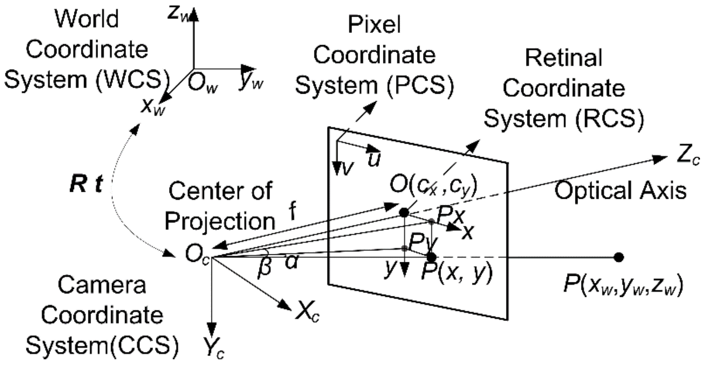


Figure 3. Schematic diagram of image sensor's imaging model.

Here o is the origin of the RCS and o' is the origin of the PCS. o is referred to be the principal point, which is at the intersection of the image plane and the optical axis.

Usually, the defined principle point is not equivalent to the center of the imager because of machining errors. Moreover, the center of the chip is usually not on the optical axis as a result of installation errors. Hence two new parameters, u_0 and v_0, are introduced to model a possible displacement (away from the optical axis) of the center of coordinates on the projection screen.

We define the physical size of each pixel in axis x and y as d_x, d_y. Then we use a relatively simple model to represent the transformation between coordinate systems. When a coordinate point Q(x, y) in RCS is projected onto the screen at some pixel location given by (u, v), we can describe the transformation using the following equation:

$$\begin{cases} u = \dfrac{x}{d_x} + u_0 \\ v = \dfrac{y}{d_y} + v_0 \end{cases} \tag{17}$$

Given a point $P(x_w, y_w, z_w)$ of the non-cooperative satellite in WCS, when projected onto the screen, we represent its 2D location by $P(x, y)$ in RCS. Define line of sight O_cP, and O_cpx is projection vector in $O_cX_cZ_c$ plane of the line of sight O_cP. Define α as the position angle, which is the angle between O_cpx and O_cZ_c. O_cpy is projection vector in $O_cY_cZ_c$ plane of the line of sight O_cP. Define β is the angle of site, which is the angle between O_cpy and O_cP:

$$\begin{cases} \alpha = \arctan\dfrac{(u-u_0)d_x}{f} \\ \beta = \arctan\dfrac{(v-v_0)d_y}{f} \end{cases} \tag{18}$$

Given the relation that maps the point $P(x_w, y_w, z_w)$ in WCS on the projection screen represented as coordinates $P(u, v)$, we can calculate α,β by Equation (18) according to the research described in [29], where d_x, d_y can be cato the physical size of the CCD and picture resolution. They can also be acquired from the calibration results.lculated according

3. Visual Servoing Controller

Define α as the non-cooperative satellite's angle of position in the Operational Robot's field of view (FOV), β is the angle of the site. We design a visual servoing controller using a PD controller with a preset dead zone. Some similar works can be seen in [30,31]. Then we design the control law as follows:

$$\begin{cases} F_{y_r} = & -(K_{p\alpha} + K_{d\alpha}s)\alpha \qquad While\,|\alpha| & \geqslant \alpha_0 \\ F_{z_r} = & -(K_{p\beta} + K_{d\beta}s)\beta \qquad While\,|\beta| & \geqslant \beta_0 \end{cases} \tag{19}$$

where, F_{yr} ,F_{zr} are the control forces in direction y and z. $K_{p\alpha}$, $K_{p\beta}$ are the proportion coefficients. $K_{d\alpha}$, $K_{d\beta}$ are the damping coefficients α_0, β_0 are the dead zone of our designed controller. As described in Figure 4, α, β actually reflect the position deviation in the directions y_r and z_r.

As a result, we can control the position deviations by controlling the azimuth angles under a certain threshold range. The axes of the target's orbit coordinate $O_tX_tY_tZ_t$ are defined as follows: x-axis is in the orbital plane directed along the local horizontal, y-axis directs along the orbital normal and z-axis directs from the center of the Earth to the centroid of the space platform and completes a right-handed triad with the x-axis and y-axis. During the whole approach process, when the deviations of α, β are restricted under a certain threshold, the position deviations will gradually decrease since the distance to the Operational Robot shortens.

Three factors should be considered when selecting a proper α_0, β_0 First, they should satisfy the requirement of position control. Second, the need for frequent control should be avoided in order to save mass carrier. Furthermore, the monocular camera's FOV is worth being considering.

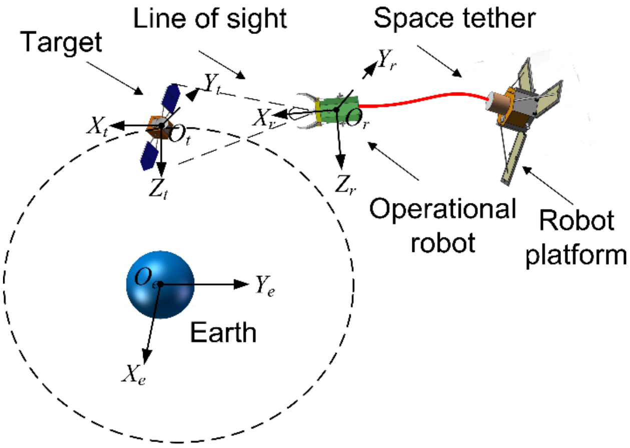


Figure 4. Schematic diagram of the TSR's approach using azimuth angles.

4. Experimental Validation

4.1. Experimental Set-up

We set up a servoing control system based on monocular vision to test our algorithm. It is composed of a control subsystem, visual perception subsystem, intelligent motion platform and a target simulator. The visual perception subsystem uses a HaiLong M200A camera (Camerasw, Changsha, China) mounted on a 1/3.2 inch (4:3) CMOS sensor (Micron Technology, Boise, ID, USA) and a Computar M1614 fixed focus lense with 23.4° FOV (Computar, Tokyo, Japan), 16 mm focal length. The image resolution is set as 1280 pixel × 960 pixel. A HP workstation (the CPU is a Xeon 5130 2.0 GHz while its RAM is 2G) (Hewlett-Packard Development Company, Palo Alto, CA, USA) is used to obtain the experimental results. We then implemented our algorithm in the Windows XP system using VS2010 IDE installed OpenCV 2.4.3 libraries.

4.2. Design of Experiments

In our laboratory, a series of semi-physical experiments were designed to test our method's performance. The experimental system used in our lab is shown in Figure 5.

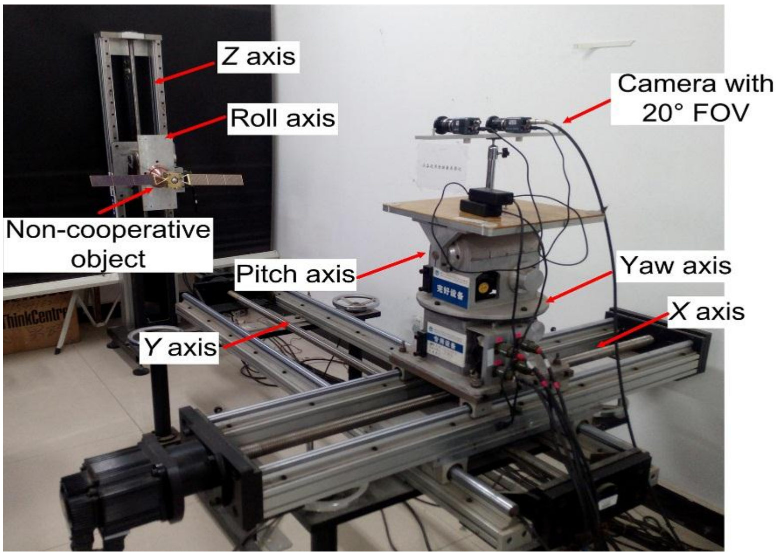


Figure 5. Six DOF motion platform and target model.

To simulate a non-cooperative target, we use a Chang'e-II satellite model, whose main body is a 70 × 70 × 60 mm^3 cube. The span of its solar panel is about 300 mm long. A six DOF motion platform is used to simulate their relative movement, which consists of a two DOF slipway, two DOF (pitch and yaw) revolving table, and two DOF lifting and roll platforms. The six axes can be controlled separately or together by the PEWIN32RO software or a C++ program, including acceleration, velocity, and amplitude parameters. The system performance is summarized in Table 1.

Table 1. Performance of the 6 DOF moving platform.

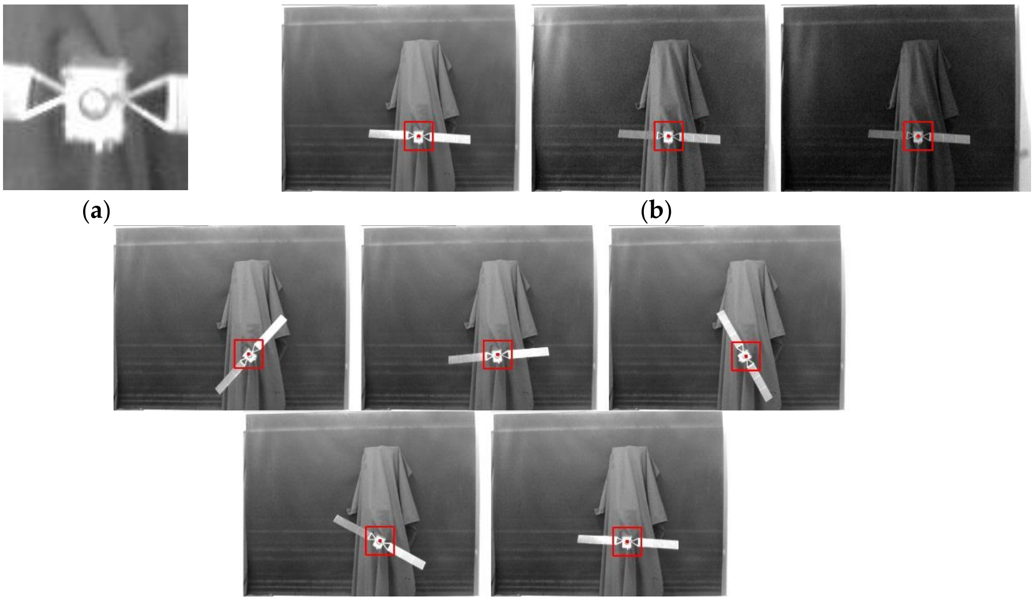
Axis	**Route (mm)**	**Velocity (m/s)**	**Acceleration (m/s^2)**
X axis	0–1000	±0.01–1	±0.01–1
Y axis	0–1500	±0.01–1	±0.01–1
Z axis	0–1200	±0.01–1	±0.01–1
Axis	Amplitude (°)	Angle Velocity (°/s)	Angle Acceleration (°/s^2)
pitch axis	±45	±0.01~10	±0.01~10
yaw axis	±90	±0.01~10	±0.01~10
roll axis	±90	±0.01~10	±0.01~10

The lights system, background simulation and any other support system can be separately or jointly controlled, too. We then use the system for simulating several curved approach paths to test our method in detail. In the experiments, we also design some challenge scenarios.

4.3. Results and Discussion

4.3.1. Qualitative Analysis

The size of template image selected in this paper is 60 pixels × 60 pixels, as shown in Figure 6a.


(**a**) (**b**)

(**c**)

Figure 6. Qualitative tests of the proposed template matching algorithm.

Figure 6b marks the matching region under different illumination conditions. In this experiment, the lights system is controlled to simulate different illumination conditions, ranging from bright to dark. Generally, the illumination characterizes nonlinear changes. In Figure 6b, we can see our method is robust to brightness changes.

Figure 6c shows the matching results with different rotation angles. Satellites mounted on the two DOF lifting and roll platforms are controlled to rotate with different arbitrary angles. For these conditions, the template image is used as shown in Figure 6a. From the matching results drawn in the red rectangle, we can conclude that our method is robust under a certain illumination and rotation.

4.3.2. Quantitative Comparisons

To test the proposed method deeply and practically, we merged the method into the TSR's monocular visual servoing system and validated it with semi-physical experiments. Then we recorded all the matched regions that are used for obtaining azimuth angles.

Figure 7 shows the matching results by our method in the first 717 frames. In this experiment, the six DOF motion platform is controlled to pick up the camera towards the satellite model. During the approach process, there are changes in the axis of x, y, z, pitch, yaw and roll while the lights system is controlled to simulate different illumination conditions. It can be seen that the template image was matched relatively well.

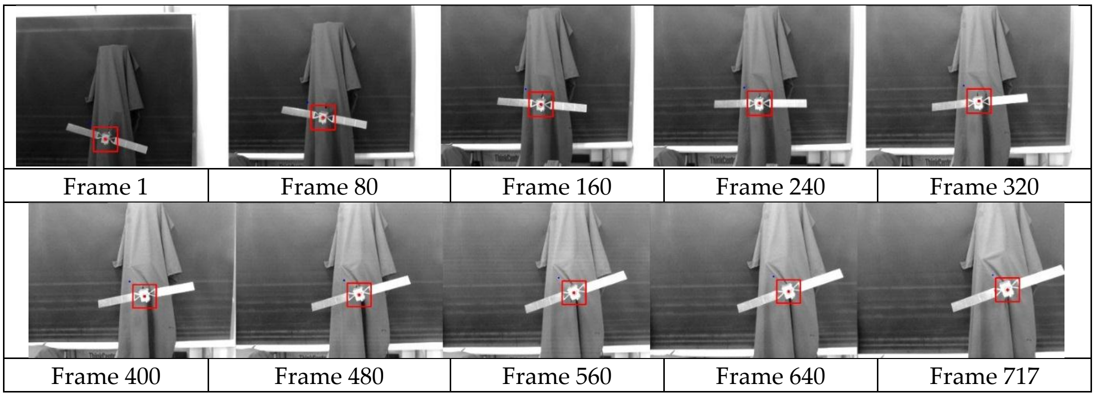
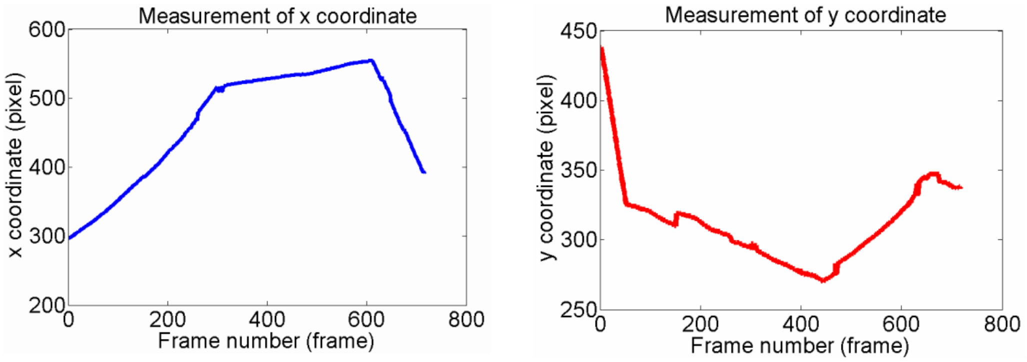


Figure 7. Tracking results of our algorithm during the first 717 frames.

Figure 8 shows the measurements of a matched target's centroid (x, y) in the first 717 frames. During the approach process, our method uses the template image as shown in Figure 6a. From Figure 8, it can found that although the original target rotates arbitrarily angles and endures different illumination, the curve of the tracking or detection results is smooth.



(**a**) (**b**)

Figure 8. Measurements of coordinate x and y in the first 717 frames. (**a**). Measurement of x coordinate; (**b**) Measurement of y coordinate.

As the CMOS is 1/3.2 inch (4:3) and the image resolution used is 1280 pixel × 960 pixel, it can be calculated the the physical size of the longer side is 6.35 mm, and the shorter one is 4.76 mm.

Thus, we can acquire dx = 6.35/1280, dy = 4.76/960. According to the description in Section 2.4, the corresponding azimuth and site angles can be calculated as shown in Figure 9.

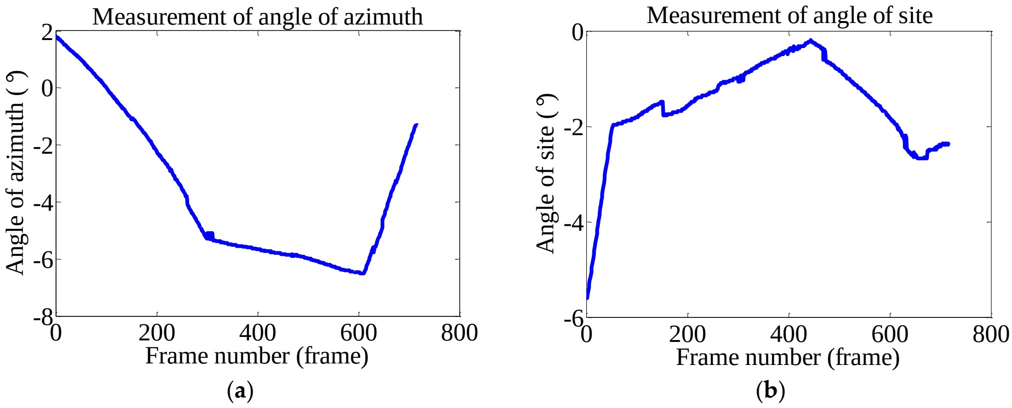


Figure 9. Measurements of angles of site and azimuth in the first 717 frames. (**a**) Measurement of the azimuth angle; (**b**) Measurement of the site angle.

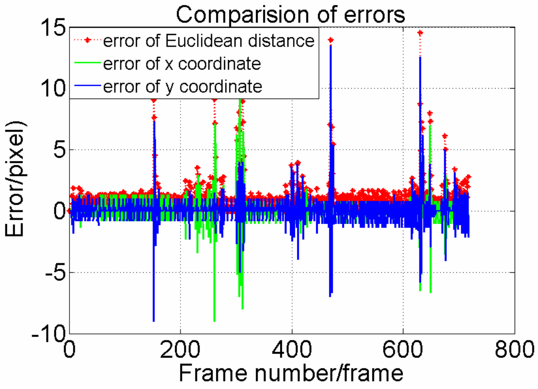


Figure 10. Comparison of prediction error per frame of the least square synthesis predictor.

We also analyze the performance of the synthesis predictor. From Figure 10, accurate error data are drawn in detail. The max absolute error of the x coordinate is 9.5 pixel and the mean absolute error is 0.75 pixel. The max absolute error of the y coordinate is 13.5 pixel and mean absolute error is 0.84 pixel. The max absolute error of the Euclidean distance is 14.57 pixel and mean absolute error is 1.28 pixel. It can be concluded that our designed predictor has good accuracy performance.

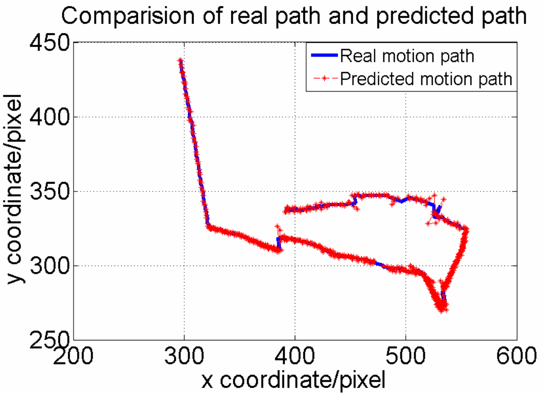
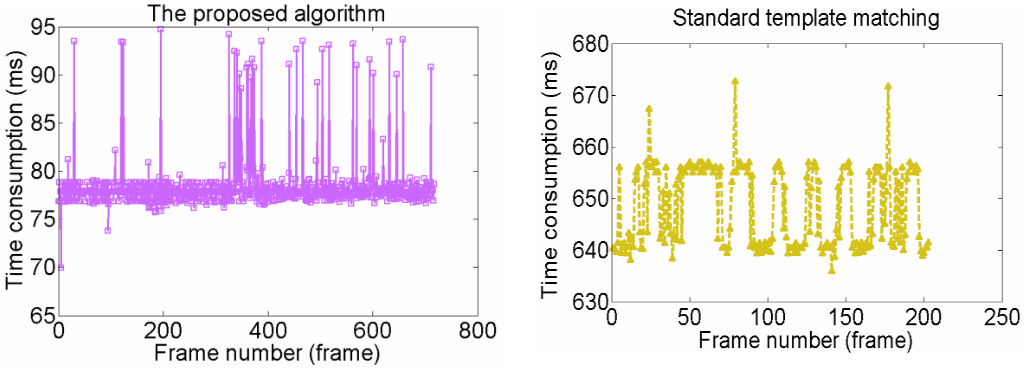


Figure 11. Comparison of path generated by our algorithm and real path.

Based on the data, we can also draw the real motion path and the predicted one of the Operational Robot during the approach process. In Figure 11, the red line marked with a triangle is the real motion path while the blue one is the predicted one. We can intuitively find that the predicted one highly coincides with the real one. Once the predictor gives the possible position, the small template image only has to search for a matching scene within a 50 pixels × 50 pixels window centered on this point. As a result, it can decrease sophisticated computation and improve the matching accuracy. In contrast, standard template matching was also used to recognize the same target. The time consumed by both processes in each frame were recorded. Figure 12a describes the time consumption in each frame of our proposed method (M1). Figure 12b describes time consumption in each frame of the standard template method (M2). From the data, we can find that M2 failed to work from frame 204, so Figure 12b only draws the curve of time consumption for the first 203 frames. We can see that our method worked well during the whole process. We can also calculate that the average time consumptions of M1 and M2 are 78.42 ms and 648.16 ms, respectively.



(**a**) (**b**)

Figure 12. Comparison of time consumption per frame between the two algorithms.

To validate our novel dynamic template matching method in a natural scene, we applied the method to deal with a video of a tracking dataset as shown in Figure 13. In this video, a boy rode a bike from the left side to the right while there was a tree in the scene. From the 13th to 41th frame, the boy on the bike was blocked by the tree.

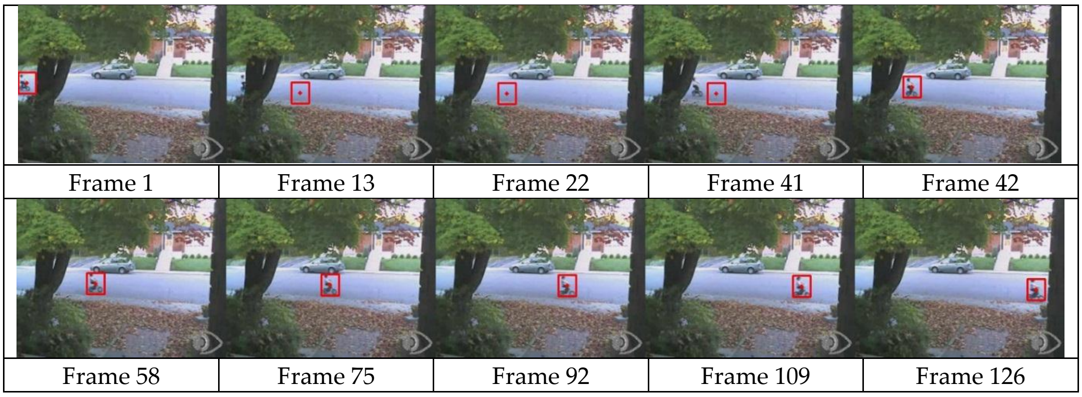


Figure 13. Tracking results of object with occlusion in natural scene.

When the target crossed behind the tree, the true target could not be found, so the matched sub-image in 13th to 41st frame is not the real target. In these frames, the similarity function is much larger than those computed in normal tracking frames, such as the 42nd to 126th. The difference $d_M > T_d$, and the searched sub-image are viewed as a mismatched one. Then the search range changes from the rectangular candidate window R_c to the whole image. Then from 42 nd frame, the occlusion disappears and $d_M < \mathrm{T}_d$. The object is tracked successfully, and least square integrated predictor works again. From the 42 nd frame on, the method narrows down the search range to the rectangular candidate window R_c again. From this frame sequence, it can be found that our method has anti-occlusio ability.

5. Conclusions

In this paper, we present a novel dynamic template matching that may be used for a TSR's monocular visual servoing scheme. To meet the robot's performance requirements in robustness, adaptability and tracking system effectiveness, our main contributions are fourfold: (1) A prototype framework of a real-time servoing control system using monocular-CMOS is proposed; (2) A similarity function named NSAD is designed; (3) An improved matching criterion by constructing a series of hollow annulus structures from square regions and counting both gray and gradient values is defined; (4) A least squares integrated predictor and an updating strategy are proposed. A large volume of experimental results of virtual, actual and robotic tests show that our proposed matching method can quickly detect assigned small template regions with excellent accuracy under various environmental conditions. In this paper, we present only a preliminary prototype of the robotic visual servoing system. Limited to our experimental conditions, it is hard to verify the method in space. Although we achieved success in our ground experiments, our goal in future works is to combine IBVS and PBVS controllers in a switched system controller and we expect to deal with the problems of keeping the observed ROI in the camera field of view. How to profoundly improve the template matching and make it be scale invariant will be addressed in our future work as well.

Acknowledgments: This research is sponsored by the National Natural Science Foundation of China (Grant No. 11272256 and 60805034) and the Doctorate Foundation of Northwestern Polytechnical University (Grant No. CX201304).

Author Contributions: All authors are responsible for the concept of the paper, the results presented and the writing. The authors have read and approved the final published manuscript. This work was mainly conducted by Jia Cai under the supervision of Panfeng Huang. Jia Cai has developed the algorithm, conducted the experiments and drafted the paper. Bin Zhang helped perform the experiments. Dongke Wang designed the visual servoing controller.

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

References

1. Xu, W.; Liang, B.; Li, B.; Xu, Y. A universal on-orbit servicing system used in the geostationary orbit. *Adv. Sp. Res.* **2011**, *48*, 95–119. [CrossRef]
2. Coleshill, E.; Oshinowo, L.; Rembala, R.; Bina, B.; Rey, D.; Sindelar, S. Dextre: Improving maintenance operations on the international space station. *Acta Astronaut.* **2009**, *64*, 869–874. [CrossRef]
3. Friend, R.B. Orbital express program summary and mission overview. In *Sensors and Systems for Space Applications II*; SPIE: San Francisco, CA, USA, 2008; pp. 1–11.
4. Nock, K.T.; Aaron, K.M.; McKnight, D. Removing orbital debris with less risk. *J. Spacecr. Rocket.* **2013**, *50*, 365–379. [CrossRef]
5. Liu, J.; Cui, N.; Shen, F.; Rong, S. Dynamics of Robotic GEostationary orbit Restorer system during deorbiting. *IEEE Aerosp. Electron. Syst. Mag.* **2014**, *29*, 36–42.
6. Flores-Abad, A.; Ma, O.; Pham, K.; Ulrich, S. A review of space robotics technologies for on-orbit servicing. *Progress Aerosp. Sci.* **2014**, *68*, 1–26. [CrossRef]
7. Huang, P.; Cai, J.; Meng, Z.J.; Hu, Z.H.; Wang, D.K. Novel Method of Monocular Real-Time Feature Point Tracking for Tethered Space Robots. *J. Aerosp. Eng.* **2013**. [CrossRef]
8. Crétual, A.; Chaumette, F. Application of motion-based visual servoing to target tracking. *Int. J. Robot. Res.* **2001**, *20*, 878–890. [CrossRef]
9. Petit, A.; Marchand, E.; Kanani, K. Vision-based space autonomous rendezvous: A case study. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011), San Francisco, CA, USA, 25–30 September 2011; pp. 619–624.

10. Larouche, B.P.; Zhu, Z.H. Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control. *Auton. Robot.* **2014**, *37*, 157–167. [CrossRef]
11. Wang, C.; Dong, Z.; Yin, H.; Gao, Y. Research Summarizing of On-orbit Capture Technology for Space Target. *J. Acad. Equip.* **2013**, *24*, 63–66.
12. Du, X.; Liang, B.; Xu, W.; Qiu, Y. Pose measurement of large non-cooperative satellite based on collaborative cameras. *Acta Astronaut.* **2011**, *68*, 2047–2065. [CrossRef]
13. Yilmaz, A.; Javed, O.; Shah, M. Object tracking: A survey. *Acm Comput. Surv. (CSUR)* **2006**, *38*, 1–45. [CrossRef]
14. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 2nd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2002.
15. Ivan, I.A.; Ardeleanu, M.; Laurent, G.J. High dynamics and precision optical measurement using a position sensitive detector (PSD) in reflection-mode: Application to 2D object tracking over a smart surface. *Sensors* **2012**, *12*, 16771–16784. [CrossRef] [PubMed]
16. Máthé, K.; Buşoniu, L. Vision and control for UAVs: A survey of general methods and of inexpensive platforms for infrastructure inspection. *Sensors* **2015**, *15*, 14887–14916. [CrossRef] [PubMed]
17. Hutchinson, S.; Hager, G.D.; Corke, P. A tutorial on visual servo control. *IEEE Trans. Robot. Autom.* **1996**, *12*, 651–670. [CrossRef]
18. Kragic, D.; Christensen, H.I. Survey on visual servoing for manipulation. *Comput. Vis. Act. Percept. Lab. Fiskartorpsv* **2002**, *15*, 6–9.
19. Herzog, A.; Pastor, P.; Kalakrishnan, M.; Righetti, L.; Bohg, J.; Asfour, T.; Schaal, S. Learning of grasp selection based on shape-templates. *Auton. Robot.* **2014**, *36*, 51–65. [CrossRef]
20. Li, H.; Duan, H.B.; Zhang, X.Y. A novel image template matching based on particle filtering optimization. *Pattern Recognit. Lett.* **2010**, *31*, 1825–1832. [CrossRef]
21. Türkan, M.; Guillemot, C. Image prediction: Template matching *vs.* Sparse approximation. In Proceedings of the 2010 17th IEEE International Conference on Image Processing (ICIP 2010), Hong Kong, China, 26–29 September 2010; pp. 789–792.
22. You, L.; Xiu, C.B. Template Matching Algorithm Based on Edge Detection. In Proceedings of the 2011 International Symposium on Computer Science and Society (ISCCS 2011), Kota Kinabalu, Malaysia, 16–17 July 2011; pp. 7–9.
23. Lu, X.H.; Shi, Z.K. Detection and tracking control for air moving target based on dynamic template matching. *J. Electron. Meas. Instrum.* **2010**, *24*, 935–941. [CrossRef]
24. Paravati, G.; Esposito, S. Relevance-based template matching for tracking targets in FLIR imagery. *Sensor* **2014**, *14*, 14106–14130. [CrossRef] [PubMed]
25. Opromolla, R.; Fasano, G.; Rufino, G.; Grassi, M. A Model-Based 3D Template Matching Technique for Pose Acquisition of an Uncooperative Space Object. *Sensors* **2015**, *15*, 6360–6382. [CrossRef] [PubMed]
26. Cai, J.; Huang, P.F.; Wang, D.K. Novel Dynamic Template Matching of Visual Servoing for Tethered Space Robot. In Proceedings of the Fourth IEEE International Conference on Information Science and Technology (ICIST2014), Shen Zhen, China, 26–28 April 2014; pp. 388–391.
27. Rosten, E.; Porter, R.; Drummond, T. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 105–119. [CrossRef] [PubMed]
28. Xu, Y.; Yang, S.Q.; Sun, M. Application of least square filtering for target tracking. *Command Control Simul.* **2007**, *29*, 41–42.
29. Zeng, D.X.; Dong, X.R.; Li, R. Optical measurement for spacecraft relative angle. *J. Cent. South Univ.* **2007**, *38*, 1155–1158.
30. Pounds, P.E.; Bersak, D.R.; Dollar, A.M. Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control. *Auton. Robot.* **2012**, *33*, 129–142. [CrossRef]
31. Wang, D.K.; Huang, P.F.; Cai, J.; Meng, Z.J. Coordinated control of tethered space robot using mobile tether attachment point in approaching phase. *Adv. Sp. Res.* **2014**, *54*, 1077–1091. [CrossRef]