

Research Article

Vision-Based Object Recognition and Precise Localization for Space Body Control

Zeyu Shangguan ,^{1,2} Lingyu Wang,^{1,2} Jianquan Zhang,^{1,2} and Wenbo Dong ,^{1,2}

¹*Key Laboratory of Space Utilization, Chinese Academy of Sciences, Beijing 100094, China*

²*Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing 100094, China*

Correspondence should be addressed to Wenbo Dong; wbdong@csu.ac.cn

Received 29 September 2018; Accepted 2 December 2018; Published 25 March 2019

Academic Editor: Paolo Gasbarri

Copyright © 2019 Zeyu Shangguan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The space motion control is an important issue on space robot, rendezvous and docking, small satellite formation, and some on-orbit services. The motion control needs robust object detection and high-precision object localization. Among many sensing systems such as laser radar, inertia sensors, and GPS navigation, vision-based navigation is more adaptive to noncontact applications in the close distance and in high-dynamic environment. In this work, a vision-based system serving for a free-floating robot inside the spacecraft is introduced, and the method to measure space body 6-DOF position-attitude is presented. At first, the deep-learning method is applied for robust object detection in the complex background, and after the object is navigated at the close distance, the reference marker is used for more precise matching and edge detection. After the accurate coordinates are gotten in the image sequence, the object space position and attitude are calculated by the geometry method and used for fine control. The experimental results show that the recognition method based on deep-learning at a distance and marker matching in close range effectively eliminates the false target recognition and improves the precision of positioning at the same time. The testing result shows the recognition accuracy rate is 99.8% and the localization precision is far less than 1% in 1.5 meters. The high-speed camera and embedded electronic platform driven by GPU are applied for accelerating the image processing speed so that the system works at best by 70 frames per second. The contribution of this work is to introduce the deep-learning method for precision motion control and in the meanwhile ensure both the robustness and real time of the system. It aims at making such vision-based system more practicable in the real-space applications.

1. Introduction

Space programs on space robot, debris removing, rendezvous and docking, satellite formation, and other on-orbit service applications all involves the technology of moving body control [1–4]. The precondition for moving body control is to at first be acquainted with the body movement information, such as inertia, position, attitude, and velocity. Figure 1 shows several examples of on-orbit service application with vision system.

Generally, there are many techniques for measuring the relative position and attitude between two objects. Sensors such as GPS, gyroscopes, accelerometers, and star sensors are commonly used for self-navigation, and their position information is sent to each other by wireless communication.

Optical-electronic sensors such as laser radar and vision-based system may be more suitable to measure relative position and attitude when the two objects are at the close range, especially in autonomous vehicles or aircrafts [5–7]. In addition, the vision-based system based on computer vision is also widely used for object localization in industry manufacturing lines, medical instruments, and some intelligent applications. Since the vision-based measurement is of low cost and flexible to setup, vision system is increasingly applied in space body control.

Vision system also has many schemes such as monocular vision [8], stereo vision [9], and active vision with structure light [10]. Besides, the active cameras such as Flash-LIDAR's can be used to detect the unknown object [11]. For the noncooperative system localization, stereo vision

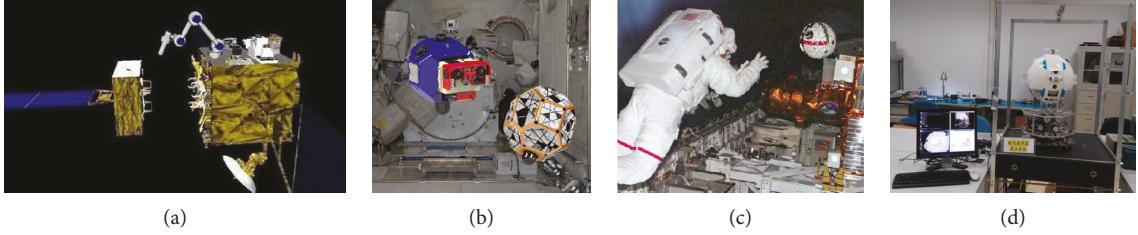


FIGURE 1: Examples of on-orbit service application with vision system. (a) On-orbit servicing, (b) SHPERE, (c) minCAM, and (d) our system on the ground.

and monocular vision [12] can both recognize the unknown objects by edge detection and feature matching. Active vision with structure light obtains the object 3D information when structure light scans the object surface, and it can not only help to recognize the object but also reconstruct the 3D information of unknown object. While monocular vision is more difficult to handle the unknown object, its precision and speed is not worse than stereo vision or active vision in applications of known target and environment.

This work is involving in an on-orbit service application. A free-floating robot can move inside the spacecraft, and the function of the robot includes routing inspection, astronaut assistant, and autonomous docking and charging. Many similar programs have been carried out in the satellite or Space Station such as the SHPEREs [13], SCAMP [14], mini AERCam [15], and Astrobee [16], which are shown in Figure 2. This kind of robot does not require space orbit control but only serves for relative movement inside the spacecraft. However, once the environment suitability permitted, it can also work out of the spacecraft. In this work [17], a vision navigation camera is configured on a robot, and another camera is fixed at the docking place to recognize this robot. These two kinds of vision system above have the same function to measure the relative position and attitude.

The main problem of the positioning system is that the complex background and light environment may influence the image recognition. Another problem is the requirement of real-time processing speed and high precision in the control system. To resolve these problems, firstly, the deep-learning method is introduced robust for object detection and localization in the image sequence. Secondly, after the object is detected, the geometry of object position and attitude calculation is solved by P4P (perspective in 4 points) method, and the explicit calculation is gotten. The embedded electronic platform driven by GPU is applied for accelerating the image processing speed.

The ground test platform is established, and its testing result indicates that our measures greatly improved the recognition accept rate so that the precision of object localization is up to 1% and the embedded platform can process the image sequence at best by 70 frames per second. The works aimed at making such vision-based system more practicable in the real dynamic environment.

2. System Design and Working Mode

The vision-based positioning system works as shown in Figure 2. The system consists of a light source, camera and

lens, and an embedded computer base on GPU and ARM. The system is tested in a scene with both simple and complex background that simulate the real-space environment. The reference marker with robust patterns of hamming code is set up both on the fixed place and on the robot, which are shown in Figure 2.

The system works in two modes as follows:

- (1) Long-Distance Navigation and Control. When the object is at a distance, the system recognizes the target from the image sequence and gets its coarse distance and position. This can be realized by identifying the position of the object located in the image and the image size d , then the coarse distance of the object can be known according to the real size of it. The robot is controlled close to the object under the help of a navigation system. In this working mode, we only estimate the position and distance of the object, without calculating its accurate attitudes
- (2) Close Docking and Precise Control. When the two objects are less than 1.5 meters apart, the camera seeks the known marker on the target and gets its accurate information to implement the precise control

3. Object Detection

According to the system working modes, the object detection process includes two parts: one is coarse recognition to find the object at a distance in the complex scenes, and the other is fine localization for calculating the accurate spatial position at close range.

3.1. Object Detection by Deep Learning. The traditional method to localize the target from the image is to directly adopt the precise image matching such as by SIFT features and ORB features [18, 19], but this method does not work well to distinguish the target in a complex background and may lead to mismatch and decrease in positioning accuracy. With the development of machine learning, the deep learning shows higher robustness and effectiveness to solve the problem of target detection in complex scenes [20, 21]. In this work, we use a method of target detection based on convolution neural network.

The target detection process is as follows: firstly, prepare the training set, label the positive and negative samples, and train and get the model with this dataset. Then, the algorithm is embedded in the computer, and the software will automatically determine whether our target is in the scene.

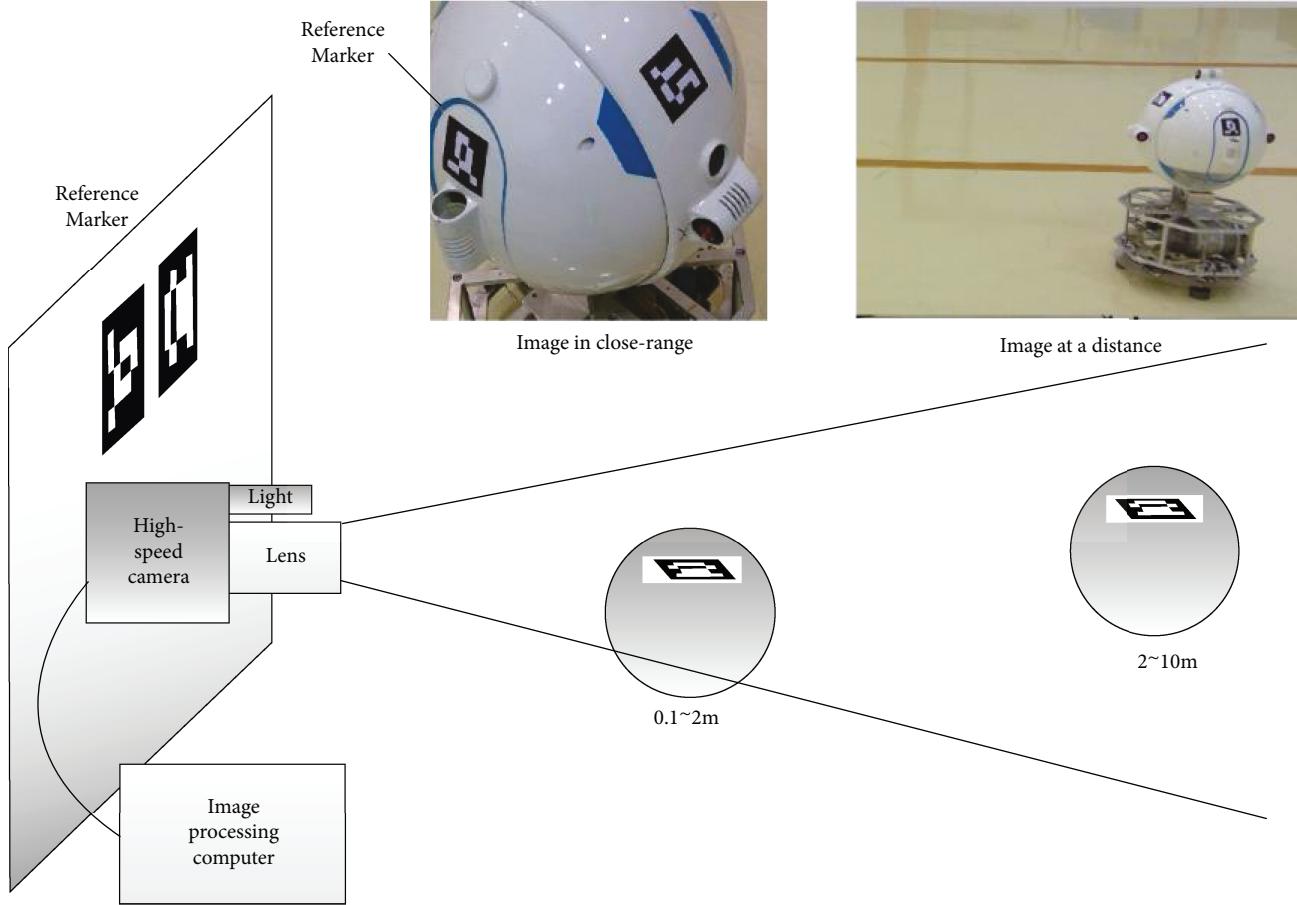


FIGURE 2: System working mode: long-distance navigation and close dock in.

3.2. Dataset Organization and Image Preprocessing. For off-line training, we select about 4000 positive samples and 9700 negative samples. It is common practice to use about 2/3 to 4/5 samples for training. The training set and test set used in this paper is 7:3. The positive samples are captured in different angles and different light intensity, and some are half-sheltered as shown in Figure 3. Negative samples without the target are also captured in the same environment, and some figures are downloaded from the internet as shown in Figure 4. Before data is imported into the network, the data need to be preprocessed. Firstly, all images are normalized to the dimension of 256×256 pixels before importing the network. Secondly, image type is converted to a certain format, so that training software can use it directly. Finally, the mean of the training set is generated to test the training accuracy.

3.3. Training Configuration. In this paper, the neural network model is divided into eight layers, five convolution layers, and three fully connected layers. In each convolution layer, the excitation function ReLU (rectified linear unit) and local response normalization LRN (Local Response Normalization) are included and then downsampling for pooling. The batch_size is set to 50, and the training sample has 9590 images, so it takes 192 iterations to complete all the samples once. The test sample has 4110 images, batch_size is 50, and

it takes 83 times to complete the test at a time. We set the snapshot to 1000, and the loss curve and accuracy curve are shown in Figure 5.

When the iterations are 768, the accuracy rate of training is 99.976% and the error rate of testing is 0.157%. The system basically reaches the steady state. And when the iterations are 30000, the model can basically meet the needs of this experiment. The training is done offline, and the time is about 60 hours in CPU or 24 hours in GPU.

3.4. Recognition Algorithm Test. To verify the reliability of the method, we use the database of 4110 samples to match, as shown in Table 1. With the alternation of ratio values, 10 groups of FPR (true positive rate) and TPR (false positive rate) are obtained. The recognition algorithm is compared with the traditional image matching method of ORB features. The former results are FPR1 and TPR1 while the latter is FPR2 and TPR2. It is often recommended that the threshold should be taken as 0.8 for image matching. In our application, we set the threshold value between 0.4 and 0.6, where the matching result was the best. If the threshold value is greater, the generalization performance will decrease.

Among the 4110 tested samples, the method with CNN has no mistake to identified negative samples as positive samples, and only 21 positive samples are not identified. The



FIGURE 3: Positive sample set in different environment.

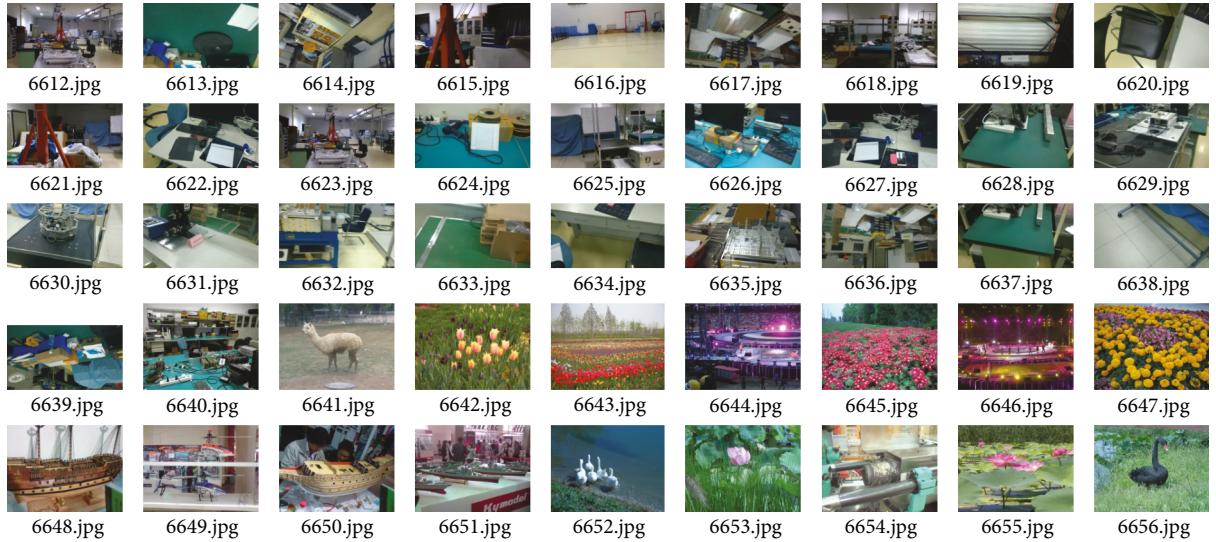


FIGURE 4: Negative sample set in different environment.

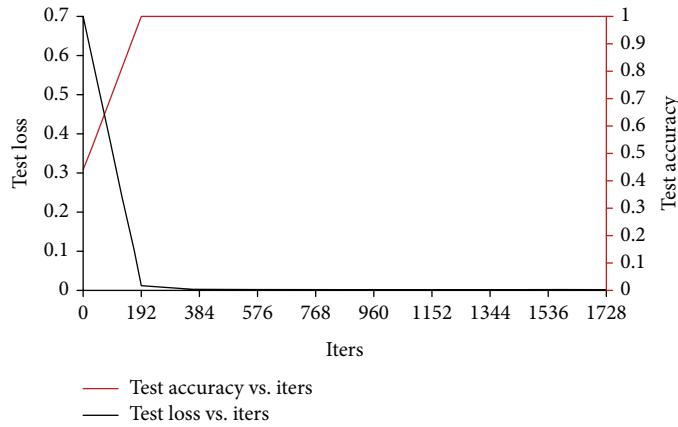


FIGURE 5: Loss curve and accuracy curve during training.

accuracy rate reached 99.847%, which is far better than the traditional object detection method. It benefits from the CNN and deep-learning technology.

3.5. Marker Recognition and Fine Localization. Marker recognition is performed when the object is less than the range of 1.5 meter and after the object has been recognized to a

TABLE 1: Image matching accuracy with different threshold.

Threshold	0.45	0.50	0.60	0.63	0.65	0.67	0.70	0.73	0.75	0.80
TPR1	23.36%	37.68%	72.28%	80.75%	85.73%	89.73%	94.19%	97.36%	98.91%	99.87%
FPR1	0	0	0	0	0	0.02%	0.06%	4.67%	18.21%	79.94%
TPR2	99.85%	99.85%	99.85%	99.85%	99.85%	99.85%	99.85%	99.85%	99.85%	99.85%
FPR2	0	0	0	0	0	0	0	0	0	0

small range. Marker detection includes the following steps: firstly, the image is normalized and converted to binary image with a certain threshold. Then, after image segmentation and contour extraction, the polygon is approximated and four vertices of polygons are found. According to the polygon position, the polygon image can be segmented and represented by 1 and 0. This binary vector is matched with hamming codes, and the number of the April Tag and its orientation is recognized in the image, as shown in Figure 6. In addition, the marker is not the only target we can use during this stage. In fact, we tried other targets that we can apply, as shown in Figure 7. However, considering the self-correcting property of the marker, the stability of it is better.

After the four vertices of the marker are confirmed, they are used for further calculation. Considering that if one marker covers 50 pixels in the 640×480 image, then the precision of one pixel leads to 1% error of localization, that is, 10 mm within the range of 1 m. For more precise localization, the coordinates of four control points are corrected to subpixel, and the error could decrease to 0.1%. There are many methods for subpixel edge detection, and we directly use the classic function of OpenCV for subpixel processing in this work. The subpixel edge detection is as shown in Figure 8. Since the subpixel detection is time-consuming, it is only used in fine operation control.

4. Position-Attitude Measurement

4.1. The Localization Algorithm. In the camera projective model, define the center of the marker as (X_W, Y_W, Z_W) in the marker coordinate system and (X_C, Y_C, Z_C) in the camera coordinate,

Here, the translation vector is $T = (T_X, T_Y, T_Z)^T$ and the rotation matrix is

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}. \quad (1)$$

The translation vector T can represent the three-dimensional position of the target in the camera coordinate system, and the rotation matrix R can characterize the attitude of the target in the camera coordinate system. Support P_0 is the original position of the moving object. The position measurement question can be shown as the below images: the question to calculate the robot's position and attitude relative

to the camera. Here is the equivalent to resolve the matrix rotation matrix R and translation vector T in

$$\begin{pmatrix} X_C \\ Y_C \\ Z_C \end{pmatrix} = R \begin{pmatrix} X_W \\ Y_W \\ Z_W \end{pmatrix} + T. \quad (2)$$

Let P_0, P_1, P_2, P_3 as four coordinates of the marker vertices in the space and q_0, q_1, q_2, q_3 as their projection coordinates in the camera plane, which is shown in Figure 9. Suppose that $c, P_0P_1//P_2P_3, |P_0P_1| = d_1, |P_0P_3| = d_2$, and the camera intrinsic parameters are known. The question to resolve the R and T is a classical P4P problem [22].

Define the control points in the camera coordinate as $q_0(x_0, y_0, f), q_1(x_1, y_1, f)$, and $q_2(x_2, y_2, f)$, where f is the camera focus, and define the optical center as O_C , the plane π_1 is formed by O_C and q_0, q_1 . The normal vector of π_1 is $N_1 = (n_x, n_y, n_z)^T$, which can be calculated by linear equation of q_0, q_1 and camera intrinsic parameter.

Let $k_0 = |O_C P_0| / |O_C q_0|, k_1 = |O_C P_1| / |O_C q_1|$.

Since vector $P_0 P_1$ is perpendicular to vector N_1 , there is

$$(k_1 x_1 - k_0 x_0) \cdot n_x + (k_1 y_1 - k_0 y_0) \cdot n_y + (k_1 f - k_0 f) \cdot n_z = 0, \quad (3)$$

$$\sqrt{(k_1 x_1 - k_0 x_0)^2 + (k_1 y_1 - k_0 y_0)^2 + (k_1 f - k_0 f)^2} = d_1. \quad (4)$$

From equations (3)~(4), k_0 and k_1 can be solved, and correspondingly P_0 and P_1 is solved as $(x_{p0}, y_{p0}, z_{p0}) = (k_0 x_0, k_0 y_0, k_0 f), (x_{p1}, y_{p1}, z_{p1}) = (k_1 x_1, k_1 y_1, k_1 f)$. If we take P_0 as the origin point, then the displacement is $T = (x_{p0}, y_{p0}, z_{p0})^T$.

The X_W axis in the camera coordinate is $(x_{p1} - x_{p0}, y_{p1} - y_{p0}, z_{p1} - z_{p0})$ and is normalized as (r_{11}, r_{21}, r_{31}) , which is the first column of rotation matrix R .

The Y_W axis in the camera coordinate is $(x_{p3} - x_{p0}, y_{p3} - y_{p0}, z_{p3} - z_{p0})$ and is normalized as (r_{12}, r_{22}, r_{32}) , which is the second column of rotation matrix R .

The Z_W axis in the camera coordinate (r_{13}, r_{23}, r_{33}) can be calculated by the cross multiplication as

$$(r_{13}, r_{23}, r_{33}) = (r_{11}, r_{21}, r_{31}) \times (r_{12}, r_{22}, r_{32}). \quad (5)$$

It is the last column of R , and by now the rotation matrix R has been solved.

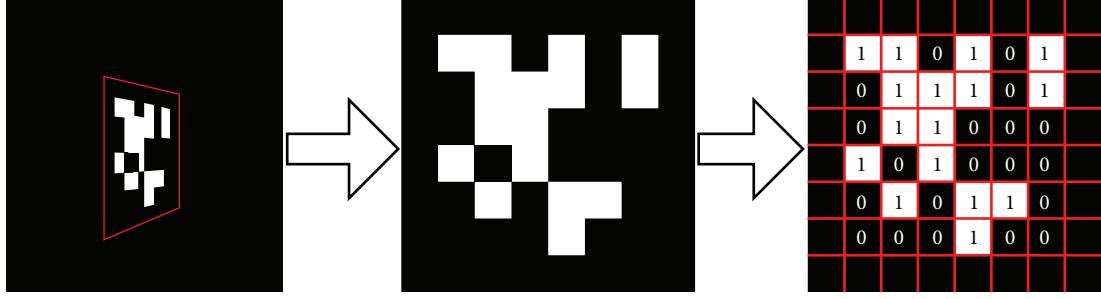


FIGURE 6: Reference marker recognition and decoding process.

5. Precision and Speed Testing

5.1. Experiment Platform

5.1.1. Experiment Design. A platform is set up to validate the precision of the system as shown in Figure 10. The reference marker is fixed on a 6-D precision displacement table, and the camera is installed on the other mechanic table. We record the displacement of the 6-DOF table as the real value, and the calculated displacement by vision system as the test value. The difference between the real value and the test value represents the precision of the system.

When the mechanical platform is moving, the computer software samples the image sequence, calculates the accurate position and attitude information of each picture, and draws the moving trajectory in the real time, as shown in Figure 11. The red square indicates the marker in camera coordinate. To reflect the direction of the marker, the upper left corner of it is defined as the first corner when it is standing and the other three corners in the counterclockwise order.

5.1.2. Camera Calibration. Before the setup of the camera, the camera should be calibrated offline at first. The camera intrinsic parameter commonly is described as

$$A = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (6)$$

where $\alpha_x = f/dX$ is the normalized focus of u axis, $\alpha_y = f/dY$ is of v axis, γ is the distortion factor, and u_0, v_0 is the image coordinate of optical center. These factors only depend on the camera itself and can be calibrated in advance.

The method of camera calibration is a commonly used operation. According the theory of Zhang [23], we only need one 2D checkerboard marker and a picture of it in various angle and position, then the inner parameters can be obtained by the least-square method.

In this work, we use a 5 mm lens and a camera with 640*480 gray in 1/3 inch; the camera parameters are calibrated as Table 2.

According to Table 2, the normalized focus of u and v axes is $(\alpha_x, \alpha_y) = [1338.84, 1345.02]$, the image coordinate of the optical center is $(u_0, v_0) = [618.15, 475.73]$, the



FIGURE 7: Various targets we can choose.

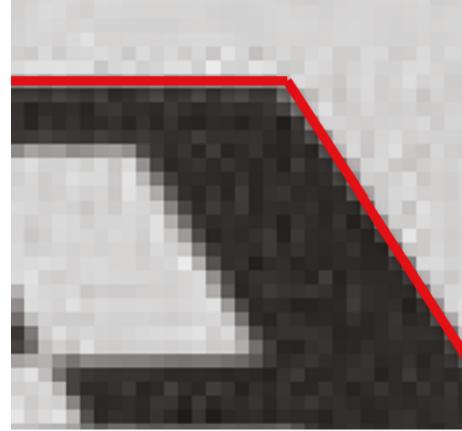


FIGURE 8: Schematic picture of subpixel edge detection.

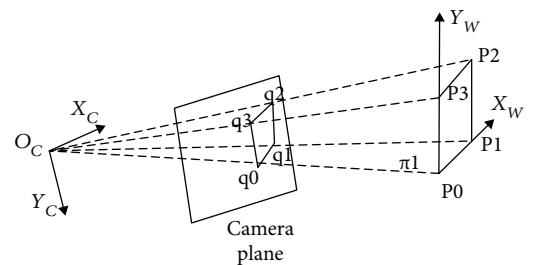


FIGURE 9: Three-dimensional pose solution.

distortion factor of u and v axes is $\gamma = 0$, and the radial distortion parameter is $k = [-0.1116, 0.1242]$.

5.2. Precision Test Results

5.2.1. Single-Dimension Precision Test. Move the mechanical table, respectively, in $X \setminus Y \setminus Z$ and $R_x \setminus R_y \setminus R_z$ axes, and test whether the calculated value is consistent with the real value.

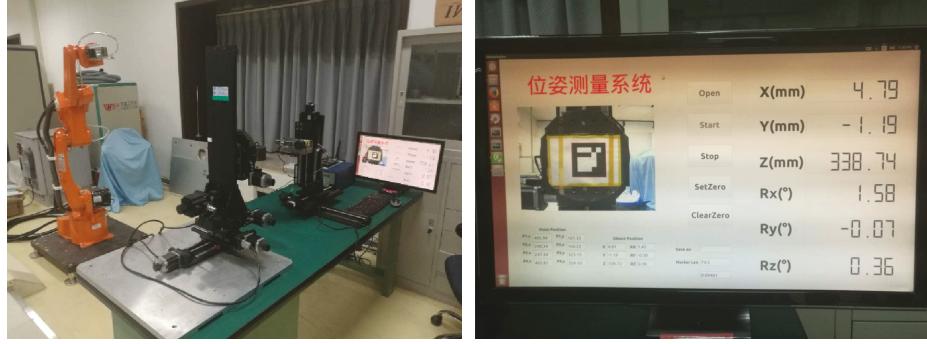


FIGURE 10: Precision experiment table.

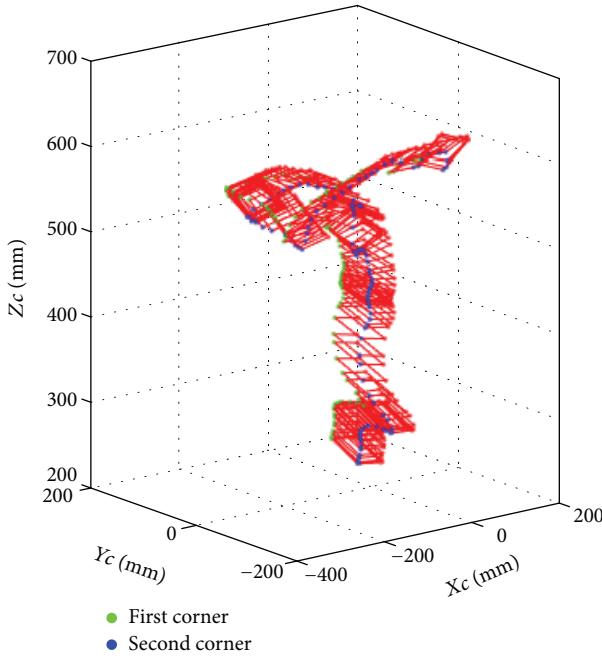


FIGURE 11: Real-time trajectory of the marker.

TABLE 2: Intrinsic parameter result of camera calibration.

α_x/pixels	α_y/pixels	γ	u_0/pixels	v_0/pixels	k_1	k_2
1338.84	1345.02	0	618.15	475.73	-0.1116	0.1242

Figure 12 is the test results in Z and R_yR_z , and the error is less than 1%, except for some points in the side face of R_y .

5.2.2. Multidimension Decoupling Test. Since the measurement of multiple degrees of freedom may have the coupling and its error model is difficult to analysis, the decoupling influence must be given by an actual test.

The corresponding experimental schematic diagram of the multidimension decoupling test is as shown in Figure 13, and Figure 14 is the curve when the object moves in a rectangle of 40×60 cm at a surface. At the same time, we record the value change of Z axis and R_z ; the results show the

decoupling with Z -XY and R_z -XY is, respectively, less than 1 mm and 2 degrees. This error may be caused by the camera calibration or the position calculation or other system error.

5.3. Real-Time Speed Test Results. For real-time operation, we choose high-speed camera with low exposure time and use a GPU embedded platform to implement and speed up the image processing algorithm. Here, the camera is from Basler gc300 (640×480 and 300 fps), and GPU is the Tegra TX1 of NVIDIA company. These industry components are for ground test and may be enhanced for space environment.

GPU performs well on big-size images and can be accelerated for image processing and CNN applications [24]. FPGA performs well on small-size images and parallel processing, and the heterogeneous computing with GPU and FPGA will be more powerful for real-time applications. The information below is only for the GPU as the processor individually.

The vision-based system works with the following steps:

- (1) Image sequence sampling
- (2) Image normalization or image preprocessing
- (3) Image recognition and matching
- (4) Position and attitude calculation

Among them, the image sequence sampling is mainly related with the camera performance, and this step costs the most time. For image processing, the processing time depends on the working modes. The image recognition by CNN is a little time-consuming, since it will seek and match the feature in the whole image. When the objects are less than 1.5 meters, the recognition algorithm is much faster, since the reference marker feature is matched in a small range and the image has been converted to binary image. The time is shown in the Table 3.

As a conclusion in this chapter, ground experiment shows that the system sampling and the fasted processing speed can get up to 70 Hz in docking mode and 30 Hz in navigation mode, which has been able to satisfy most demand of vision-based control system. Actually, there are more strategies to improve the system speed, such as using the image tracking method to reduce the seeking size of

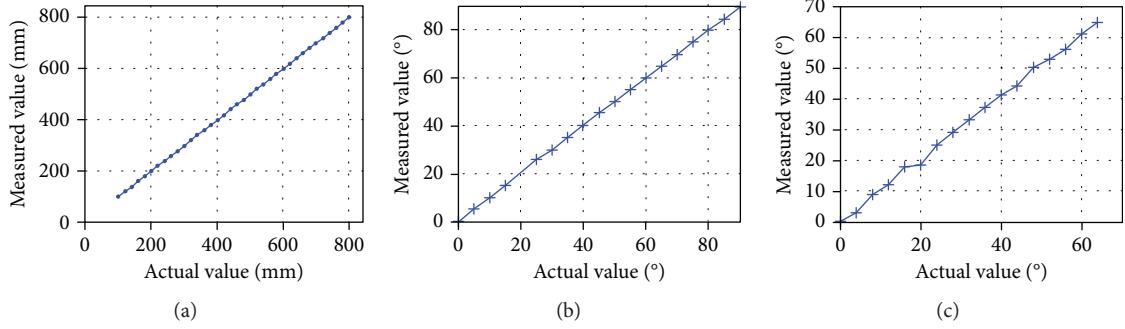


FIGURE 12: The calculated value when the object moves in different dimension. (a) Move in Z axis, (b) move in SZ axis, and (c) Move in SY axis.

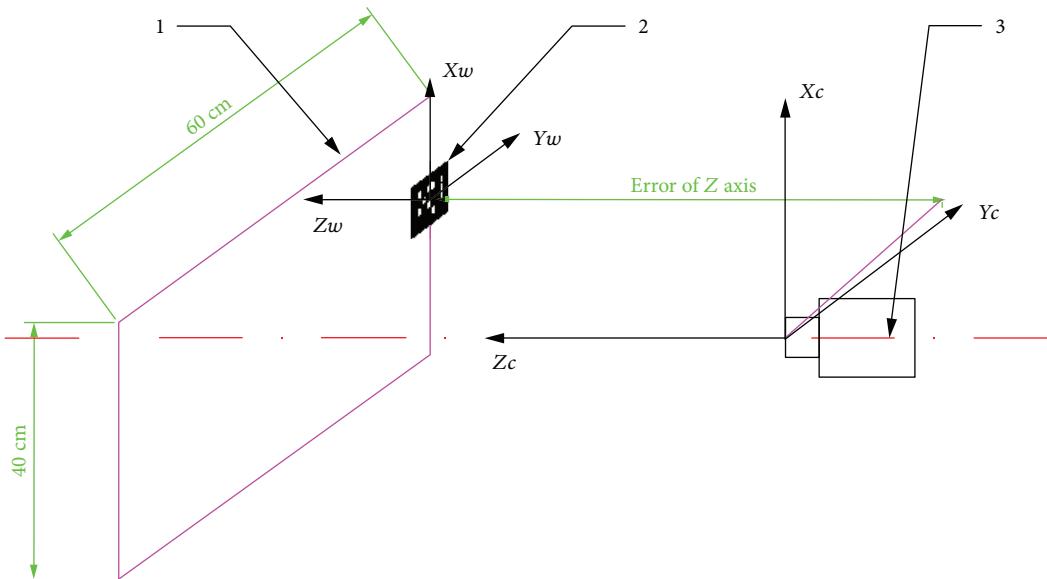


FIGURE 13: The corresponding experimental schematic diagram of the multidimension decoupling test. 1: the moving trajectory, 2: marker, and 3: camera.

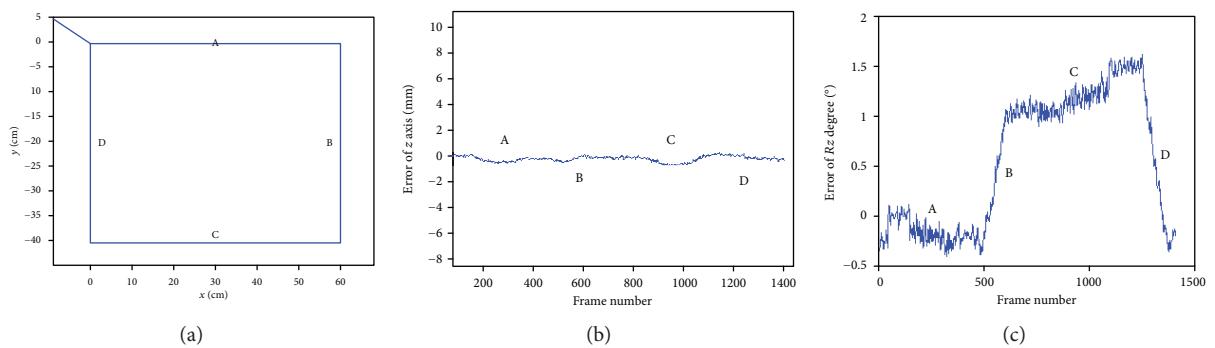


FIGURE 14: The coupling value when the object moves in a X-Y plane. (a) The XY curve, (b) Z-XY curve, and (c) R_z -XY curve.

image, or optimizing GPU program codes in CUDA accelerator, that will be discussed in the future.

6. Conclusions

A vision-based system that is mainly for space body 6-DOF position-attitude measurement and control is introduced in

this paper. The configuration of this system, the image processing and object detection algorithms, and the position-attitude measurement formula are given. Our work mainly focuses on the practical problems for the system. Firstly, we bring forward CNN with deep-learning method for object detection and get 99.8% accuracy rate. Secondly, we resolve the P4P questions for object position and attitude

TABLE 3: Time consuming for object detection and localization.

Time consuming (ms)	GPU	CPU
Image sampling and buffering	3 ms for 1 frame	3 ms for 1 frame
Image preprocessing	0.5 ms	2 ms
Object detection (navigation mode)	~30 ms	~100 ms
Object tracking (docking mode)	~10 ms	~14 ms
Coordinates calculating	<0.5 ms	<0.5 ms
Summary	Navigation mode 34 ms (~30 Hz) Docking mode 14 ms (~70 Hz)	Navigation mode 105 ms (~10 Hz) Docking mode 19.5 ms (~20 Hz)

problem and test it on the ground. The results show the precision is far less than 1% of range. The third is we use high-speed camera and GPU as the processor and accelerate the system to nearly 70 frames per second. The computer technology's fast development such as GPU and deep-learning brings us great benefits for object detection applications. In the future, we will continue to optimize algorithms and reduce the time consuming and ensure this kind of vision-based system more robust and faster and to be used in more space applications.

Data Availability

The [code] data used to support the findings of this study have been deposited in the (Scientific Data's List of Recommended Repositories) repository (DOI: 10.6084/m9.figshare.7149974).

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The authors would like to thank Tianyu Wang, Bin Yu, Hui Zhou, Xiaopeng Su, and Ting Wang for their help on this project. This work is funded by the space utilization system of Chinese Manned Spaceflight Program.

References

- [1] 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.
- [2] P. Acquatella, "Development of automation & robotics in space exploration," in *Proceedings of the AIAA SPACE 2009 Conference & Exposition*, pp. 1–7, Pasadena, CA, USA, June 2006.
- [3] 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.
- [4] S. I. Nishida, S. Kawamoto, Y. Okawa, F. Terui, and S. Kitamura, "Space debris removal system using a small satellite," *Acta Astronautica*, vol. 65, no. 1-2, pp. 95–102, 2009.
- [5] S. Ruel, T. Luu, and A. Berube, "Space shuttle testing of the TriDAR 3D rendezvous and docking sensor," *Journal of Field Robotics*, vol. 29, no. 4, pp. 535–553, 2012.
- [6] J. Galante, J. Van Eepoel, M. Strube et al., "Pose measurement performance of the argon relative navigation sensor suite in simulated flight conditions," in *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, USA, August 2012.
- [7] B. E. Tweddle and A. Saenz-Otero, "Relative computer vision-based navigation for small inspection spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 5, pp. 969–978, 2015.
- [8] R. Volpe, M. Sabatini, and G. B. Palmerini, "Pose and shape reconstruction of a noncooperative spacecraft using camera and range measurements," *International Journal of Aerospace Engineering*, vol. 2017, Article ID 4535316, 13 pages, 2017.
- [9] F. Terui, H. Kamimura, and S. Nishida, "Motion estimation to a failed satellite on orbit using stereo vision and 3D model matching," in *2006 9th International Conference on Control, Automation, Robotics and Vision*, Singapore, Singapore, December 2006.
- [10] S. Ruel, C. English, M. Anctil, and P. Church, "³D LASSO: real-time pose estimation from 3D data for autonomous satellite servicing," in *Proceeding of The 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space - iSAIRAS*, Munich, Germany, September 2005.
- [11] J. L. Sell, A. Rhodes, J. Woods, and J. A. Christian, "Pose performance of LIDAR-based navigation for satellite servicing," in *AIAA/AAS Astrodynamics Specialist Conference*, p. 4360, San Diego, CA, USA, 2014.
- [12] R. Volpe, G. B. Palmerini, and M. Sabatini, "A passive camera based determination of a non-cooperative and unknown satellite's pose and shape," *Acta Astronautica*, vol. 151, pp. 805–817, 2018.
- [13] A. S. Otero, A. Chen, D. W. Millera, and M. Hilstad, "SPHERES: development of an ISS Laboratory for formation flight and docking research," in *Proceedings, IEEE Aerospace Conference*, Big Sky, MT, USA, USA, March 2002.
- [14] G. A. Dorais and K. Nicewarner, "Adjustably autonomous multi-agent plan execution with an internal spacecraft free-flying robot prototype," in *The ICAPS Workshop on Plan Execution*, NASA Ames Research Center, Moffett Field, CA, USA, 2003.
- [15] J. Wagenknecht, S. Fredrickson, T. Manning, and B. Jones, "Design, development and testing of the miniature autonomous extravehicular robotic camera (Mini AERCam) guidance, navigation and control system," *26th Annual AAS Guidance and Control Conference*, 2003, pp. 187–205, Breckenridge, CO, USA, February 2003.
- [16] M. Bualat, J. Barlow, T. Fong, C. Provencher, and T. Smith, "AstroBee: developing a free-flying robot for the international space station," in *AIAA SPACE 2015 Conference and Exposition*, Pasadena, CA, USA, 2015.

- [17] Y. Zhang and W. Dong, *Control System Design for Micro Satellite In-Cabin Based on Visual Navigation*, IAC, 2015.
- [18] J. Molnár, D. Chetverikov, and S. Fazekas, “Illumination-robust variational optical flow using cross-correlation,” *Computer Vision and Image Understanding*, vol. 114, no. 10, pp. 1104–1114, 2010.
- [19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: an efficient alternative to SIFT or SURF,” in *2011 International Conference on Computer Vision*, pp. 2564–2571, Barcelona, Spain, November 2011.
- [20] Y. Jia, E. Shelhamer, J. Donahue et al., “Caffe: convolutional architecture for fast feature embedding,” in *Proceeding MM '14 Proceedings of the 22nd ACM international conference on Multimedia*, Orlando, Florida, USA, November 2014.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [22] L. Yueping and J. Nan, “Comparison and evaluation of PnP algorithms of monocular vision,” in *Proceedings Volume 9675, AOPC 2015: Image Processing and Analysis*, Beijing, China, October 2015.
- [23] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [24] M. Malik, F. Farahmand, P. Otto et al., “Architecture exploration for energy-efficient embedded vision applications: from general purpose processor to domain specific accelerator,” in *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 559–564, Pittsburgh, PA, USA, July 2016.

