Vision Tracking Algorithm for Augmented Reality System of Teleoperation Mobile Robots

Dongpu Zhang*
Institute of Manned Space System Engineering
China Academy of Space Technology
Beijing, China
zhangdongpu@163.com

Lin Tian
Institute of Manned Space System Engineering
China Academy of Space Technology
Beijing, China

Kewu Huang
Institute of Manned Space System Engineering
China Academy of Space Technology
Beijing, China

Jiwu Wang
School of Mechanical, Electronic and Control Engineering
Beijing Jiaotong University
Beijing, China

***Abstract*—Teleoperation mobile robot is an effective solution to the operation in unfamiliar, dangerous or unreachable environment in the deep space. With Augmented Reality(AR), teleoperation mobile robots can reach high accuracy and efficiency. To get perfect AR effect, the feature extraction and matching algorithm are achieved based on reconstruction model and k-d tree. The pose estimation used LM algorithm is discussed. A new vision tracking algorithm is built-up based on 3D reconstruction model for AR teleoperation mobile robot. With experiments, it can proved that the new vision tracking algorithm can meet the requirement of teleoperation mobile robot systems in manned deep space exploration.**

Keywords—augmented reality, teleoperation mobile robots, 3D reconstruction model, vision tracking algorithm

I. Introduction

Teleoperation robots are intelligent ones that can complete scientific tasks such as long-distance patrol detection, sampling analysis, material handling under remote control [1]. The operators monitors or controls remote robots to finish various tasks in unreachable or dangerous environments. In unfamiliar, dangerous or inappropriate environments, it is imperative to use teleoperation robots to engage in dangerous work, such as landing on the moon, asteroids or other unknown objects, instead of astronauts so as to avoid human being harmed during deep space exploration. For accurate teleoperation of robots, it is necessary to obtain the surface environment of the unknown object where the robot is located and judge the range and distance of its movement according to the environment. Because the unknown object environment is unfamiliar and complex, the two-dimensional (2D) model is not sufficient to express the true situation of the environment, so that accurate judgment of the distance cannot be realized. In previous works, Metashape software is used to reconstruct the three-dimensional (3D) environment model successfully based on the 2D images collected, which lays a technical foundation for the subsequent three-dimensional environment and models reconstructed in deep space exploration teleoperation tasks [2].

Augmented reality(AR) is an important branch of virtual reality(VR) which enhance the real world seen by users [3-4]. High reliability and accuracy required by remote teleoperation can be obtained if virtual environment is integrated with the real environment by combining augmented reality system with teleoperation robots [5-6]. In order to solve the problems of poor accuracy and inefficiency in remote operation of robot, augmented reality system for simulation control of a teleoperation mobile robot is designed and developed based on ARToolKit in our former work, which builds an augmented reality control platform on the remote control of robot, simulates the movement of robot in real environment, controls the tracking control command of analog robot, and makes the remote teleoperation efficiency improved. In order to achieve the perfect combination of virtual and real world, ultimately meets the requirements of man-machine collaborative operation tasks such as on-orbit augmented reality teleoperation for unknown celestial environment, a real-time tracking algorithm based on the reconstruction model is needed, which can get the location information, illumination information and occlusion information of some key parts of the real scene in complex large scenes of an open environment on unknown objects in real time, quickly and accurately.

II. Algorithms and Experiments

A. Principle

The principle of the real-time tracking algorithm based on the reconstructed model is tracking in real time with the parameters and information of the reconstructed 3D models [7]. The useful information of a priori model includes the original images for reconstruction, the matching relationship between the 2D feature points of the reconstructed image and the 3D points of the model, etc. Due to the usage of information related to the reconstructed 3D model, this algorithm can effectively avoid the impact of external conditions, such as changes in lighting, or changes in scenes. The basic steps of the real-time tracking algorithm based on the reconstruction model are as follow.

First, the k-d tree of the model is built offline according to the 3D reconstruction model of the target and its corresponding information. Then, for the actual image sequences, the k-d tree established before is searched quickly and accurately to find the 2D-3D matching points by Best-Bin-First(BBF) algorithm. The 2D-3D matching points found by BBF will be the input data for the posture estimation. Next, according to the theory of computer vision and multi-view geometry, the position and posture of the current frame camera, i.e. the P-matrix of the camera, will be

Funded by Manned Space Research Program (060601) .

978-1-7281-8025-0/20/$31.00 ©2020 IEEE

calculated by LM optimization algorithm, so that the target tracking is finally achieved. Fig.1 is a basic flowchart of a real-time tracking algorithm verification system based on the reconstructed model.

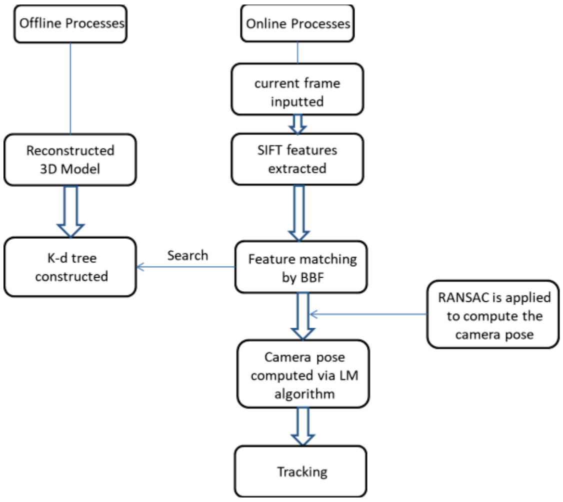

Fig. 1. Basic System Flowchart of the Algorithm

1) Offline Processes

For the tracking algorithm, it is the key point that how to establish the corresponding relationship between the 3D target and the 2D image sequences. As shown in Fig.1, in offline processes, 3D models of real world, such as such as models reconstructed in our previous work, should be constructed firstly. The priori model consisting of 3D points, which is used to reconstruct the original image and the model, and 2D points corresponding to the image, are used to build the k-d tree of the model. The k-d tree created at this time already contains the coordinates of 3D points, which can be used for feature matching in subsequent tracking processes directly.

2) Online Tracking Process

The online tracking processes mainly include the following four steps.

a) Feature point extraction

Generally, Harris feature extraction algorithm is used to extract the feature points in the current image frame and obtain the 2D image coordinates of the feature points. The advantage of Harris corner is that it is invariant to the shift of gray level, invariant to the rotation, and benefit for denoising. However, it has some limitations in tracking system due to the fact that Harris point extraction and matching is not appropriate for the situation of large viewing changes. Scale Invariant Feature Transform (SIFT) feature points are used in the real-time tracking algorithm. SIFT feature point is based on scale space, which is invariant to image scaling, rotation and even affine transformation, and it can solve the influence of illumination and environment change, which is very important for tracking system.

b) Feature matching, establishing 2D-3D matching relationship.

BBF search algorithm is used to search the k-d tree of the model to find the 3D matching points of the feature points on the image. An approximate Best-Bin-First (BBF) algorithm, based on a k-d tree search is employed[3]. A k-d tree is constructed from all SIFT features which have been extracted from the reference images. The search examines bins, or tree leaves, each containing a feature, in the order of their closest distance from the current query location. Search order is determined with a heap-based priority queue, and an approximate answer is returned after examining a predetermined number of nearest leaves. This technique finds a closest match with a high probability, and enables feature matching to run in real time.

The best candidate match for a SIFT feature is its nearest neighbor, defined as the feature with the minimum Euclidean distance between descriptor vectors. The Euclidean distance between two points is defined as (1).

$$|d - d'| = \sqrt{\sum_{i=1}^{i=kd} (d_i - d_i')^2} \tag{1}$$

Firstly, the BBF search finds the nearest and second nearest neighbor pair. Then the distances between this point and the two searched points is compared. It is considered that neither point is the matching point of the feature point if the distance between the two points is similar. Only if the distance between the two points is very large, it means that one of the two points is the matching point of the feature point. The point with small distance is the matching point of the feature point. For each image pair, this constraint can be expressed as nearest-to-second-nearest distance ratio, which can be seen in (2). $d_{nearest}$ is the nearest distance and $d_{2\text{-nearest}}$ is the second nearest distance, with the threshold, Thr, value of 0.8).

$$R = \frac{d_{nearest}}{d_{2-nearest}} \begin{cases} \geq Thr, not\ the\ match\ point \\ \leq Thr, match\ point \end{cases} \tag{2}$$

Since the k-d tree already contains the coordinates of the 3D points corresponding to each point, the 2D-3D matching relationship is established after finding the matching points of the feature points on the image.

Optimize the matching results via Random Sampling Census (RANSAC) algorithm and remove the wrong matching. According to the process of RANSAC algorithm, it can achieve a good parameter optimization result. At the same time, on this optimization result, we distinguish the outliers and interior points of samples, and the proportion of outliers can reach more than 50% in theory. For the problem of image matching, it is very effective to use RANSAC algorithm to remove mismatches.

c) The 3D model is back projected on the 2D image, and the reprojection error function is established.

Theoretically, according to the matched point pairs $(\mathbf{x_{ij}},\mathbf{X_j})$, we seek to compute world coordinates of the corresponding 3D points, calibration parameters and camera poses for each reference view. Formally, a 2D projection $\mathbf{x_{ij}}= [\mathbf{u_{ij}},\mathbf{v_{ij}},\mathbf{1}]^T$ of a 3D point $\mathbf{X_j}=[\mathbf{X_j},\mathbf{Y_j},\mathbf{Z_j},\mathbf{1}]^T$ in an image i is expressed as $\mathbf{x_{ij}}=\mathbf{P_i}\mathbf{X_j}$, $\mathbf{P_i}$ is a 3×4 camera matrix, that is, the camera pose matrix. The real-time tracking of the target will be realized only by finding out the pose of the camera in the current frame. Therefore, the ultimate goal of the tracking algorithm is to obtain the camera pose matrix $\mathbf{P_i}$ of each frame.

However, it is no doubt that the error will be quite large if we use the formula to calculate the camera matrix $\mathbf{P_i}$ of the current frame directly. In this case, the camera matrix of the current frame can be obtained by establishing the reprojection error function as the objective function of iterative optimization. Reprojection error function is defined as (3).

$$F_i = \sum_j \| P_i X_j - x_{ij} \|^2 \quad (3)$$

d) Calculate the camera pose matrix Pi of the current frame.

Taking the above reprojection error function $\mathbf{F_i}$ as the objective function, the convergence value of $\mathbf{P_i}$ is obtained by cyclic iterative method. The P matrix calculated by the plane linear algorithm is used as the initial value of the iterative algorithm. Finally, the convergence value of the iterative result is taken as the camera pose matrix $\mathbf{P_i}$ of the current frame. In the real-time tracking algorithm, Levenberg-Marquardt(LM) algorithm of nonlinear least square method is used. The convergence speed of this algorithm is faster than other algorithms, and it can meet the requirements of tracking algorithm, especially in the finite computational ability environment.

3) Levenberg-Marquardt(LM) algorithm for pose estimation

In camera parameter estimation, the simplest algorithm is dominant algorithm. However, the accuracy of the solution obtained by linear algorithm is relatively poor, because the coefficient matrix of linear equations contains various errors. In general, the optimized objective function is nonlinear function and solution of high accuracy can be achieved by nonlinear optimization method. As a result, the nonlinear optimization method is more reasonable.

According to the camera perspective projection model, the spatial feature points of the target object are projected onto the image plane to obtain the model image coordinates (U_i,V_i) of the feature points. There is a deviation between the model image coordinates (U_i,V_i) and the actual image coordinates (u_i,v_i) detected by the camera. In the tracking algorithm in this paper, this deviation is the reprojection error, as shown in (3). The optimization algorithm is to minimize this deviation, so as to obtain the parameter with the smallest deviation. The nonlinear optimization process is as follows in (4).

$$\min F(x) = \sum_{i=1}^{m} f_i^2(x), f_i^2(x) = (U_i - u_i)^2 + (V_i - v_i)^2 \quad (4)$$

To solve the minimization problem of (4), recursive search is used, so that the calculation of the minimization process of the objective function F(x) is very large. Among different general-purpose optimization algorithms, LM algorithm is the fastest one. Therefore, it is the best choice to improve the real-time performance of the tracking system. Besides, the initial value is also an important factor to the camera **P** matrix. The solution of the plane linear method is used as the initial value of the iterative optimization process.

In LM algorithm, as shown in (5),

$$d^{(k)} = -(A_k^T A_k + \alpha_k I)^{-1} A_k^T f^{(k)} \quad (5)$$

I is the n-order identity matrix and α_k is a positive real number. Obviously, if α_k=0, $\mathbf{d^{(k)}}$ was the Guass-Newton direction. If α_k was large enough, the inverse matrix $(\mathbf{A_k}^T\mathbf{A_k}+\alpha_k\mathbf{I})^{-1}$ mainly depends on $\alpha_k\mathbf{I}$. In this case, $\mathbf{d^{(k)}}$ closes to ($-\nabla F(\mathbf{x^{(k)}})$), which is the steepest descent direction of F(x) at point $\mathbf{x^{(k)}}$. Generally, if $\alpha_k \in (0, +\infty)$, the direction determined by formula (5) will be between the Guass-Newton direction and the steepest descent direction [8].

In this algorithm, the key point is how to determine the parameter α_k. Obviously, if α_k was too small, $\mathbf{d^{(k)}}$ can't be guaranteed as the descent direction. In addition, if α_k was a large value, the convergence speed will be slow down, because of $|\mathbf{d^{(k)}}| \to 0$ when $\alpha_k \to +\infty$.

With the description above, the calculation steps of LM algorithm mainly include:

a. Given initial point $\mathbf{x^{(1)}}$, initial parameter α_1>0, growth factor β>1, allowable error ε>0, F($\mathbf{x^{(1)}}$) can be calculated with $\alpha=\alpha_1$ and k=1.

b. Set α=α/β, calculate $f^{(k)}$ then A_k as below.

$$f^{(k)} = \begin{bmatrix} f_1(x^{(k)}) \\ f_2(x^{(k)}) \\ \vdots \\ f_m(x^{(k)}) \end{bmatrix} \quad (5)$$

$$A_k = \begin{bmatrix} \frac{\partial f_1(x^{(k)})}{\partial x_1} & \frac{\partial f_1(x^{(k)})}{\partial x_2} & \cdots & \frac{\partial f_1(x^{(k)})}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m(x^{(k)})}{\partial x_1} & \frac{\partial f_m(x^{(k)})}{\partial x_2} & \cdots & \frac{\partial f_m(x^{(k)})}{\partial x_n} \end{bmatrix} \quad (6)$$

c. Solve the equations as below, achieve the direction $\mathbf{d^{(k)}}$, and set $\mathbf{x^{(k+1)}} = \mathbf{x^{(k)}}+\mathbf{d^{(k)}}$.

$$(A_k^T A_k + \alpha_k I) d^{(k)} = -A_k^T f^{(k)} \quad (7)$$

d.Calculate F($\mathbf{x^{(k+1)}}$). If F($\mathbf{x^{(k+1)}}$)< F($\mathbf{x^{(k)}}$), then turn to step f, otherwise proceed to the step e.

e. If $\| \mathbf{A_k}^T f(\mathbf{x^{(k)}}) \| \le \varepsilon$, then stop the calculation and get the solution as $\mathbf{x^{(k)}}$; otherwise, set α=αβ, turn to step c.

f. If $\| \mathbf{A_k}^T f(\mathbf{x^{(k)}}) \| > \varepsilon$, stop the calculation and get the solution as $\mathbf{x^{(k+1)}}$; otherwise, set k=k+1; return to step b.

The initial parameters α_1 and factor β should be appropriate values. In our calculation they are 0.01 and 10 separately.

B. *experiment*

The verification system platform of real-time tracking algorithm based on reconstruction model is a computer with Windows XP operating system. The configuration of the computer is Intel Pentium 4, CPU with main frequency of 2.8GHz, 512MB memory and Geforce 6600 graphics card. The implementation of the algorithm adopts the third party development package, including OpenGL, OpenCV and VXL.

III. RESULTS AND DISCUSSION

A. *Process and Analysis*

The verification experiment of the algorithm mainly includes the following parts: offline establishment of k-d tree of reconstruction model; online extraction of feature points; online feature point matching; online camera pose estimation. In the verification experiment of this algorithm, the target of tracking is a stone table in an open environment. The resolution of the collected image is 640 × 480.

1) Prior reconstruction model

The prior data of this algorithm includes the reconstructed 3D model, the 3D point coordinates of the model and the corresponding 2D point coordinates. All the relevant information of the reconstruction model is saved in the text file, which is used to establish the k-d tree in the offline processes.

The reconstructed 3D model is shown in Fig.2.

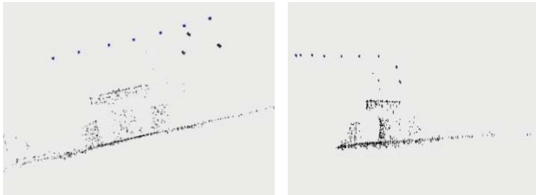

Fig. 2. Reconstruction model of stone table

Four images for reconstruction are shown in Fig.3.

Fig. 3. Original images for reconstruction

For the real-time tracking system, it is necessary to find the matching relationship between the 2D feature points and the 3D spatial points on the image for accurately calculating the camera P matrix of the current frame, and then achieving the tracking of the target. In this algorithm, k-d tree of reconstruction model is used to realize the matching of 2D-3D points, so the accuracy of reconstruction model affects the one of tracking results directly. If the error of reconstruction model is very large, it will lead to the failure of tracking.

2) Establish the K-d tree of Reconstructed Model Offline

The k-d tree based on the reconstruction model contains the coordinates of 3D points. In the following tracking process, it is quickly to search and find the matching points of the image by using the k-d tree. And then the 2D-3D matching of the image will be realized according to the information of the 3D points contained in the k-d tree.

3) Input the current frame image

Input the current frame image, as shown in Fig.4.

Fig. 4. Current frame image

4) Extract SIFT points

Extract SIFT feature points of image inputted. It takes about 5-6 seconds to extract SIFT points from 640 × 480 single image, and the number of SIFT points is about 1200. The result of extracting SIFT points from the current frame image in Fig.4 is shown in Fig.5. The number of SIFT points in the current frame image is 1221, and it takes 5.265 seconds to complete the whole process.

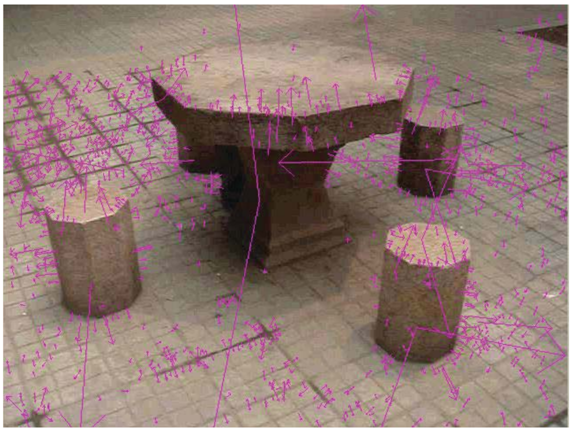

Fig. 5. SIFT points extract from current frame image

The experimental results show that more than 1000 SIFT feature points are extracted from the image, which takes about 5-6 seconds. For real-time tracking algorithm, the speed of extracting feature points is slow, which has two effects on tracking algorithm: first, the more features can be tracked, the easier it is to find feature matching between images, so as to provide accurate 2D-3D matching point pairs for camera pose estimation in the followed tracking process, which ensures the accuracy of tracking; on the other hand, the more features are tracked, the larger the slower the calculation will run because of the larger amount of data.

5) Searches 2D-3D match points searched by BBF algorithm

After SIFT feature points are extracted, the next step is to search the k-d tree of the offline reconstructed model through BBF algorithm to match the features of the image and find the coordinates of the 3D points corresponding to the 2D feature points on the image. These 2D-3D matching points will be used as input data in the camera pose estimation process. After several experiments, the time of feature matching for a single image is about 0.7 seconds. Since there are mismatches in the experimental results, these mismatches should be removed before camera pose estimation to improve the accuracy of tracking.

6) Removes mismatches by RANSAC algorithm

Better parameter optimization result can be achieved via Random Sampling Consistency (RANSAC) algorithm. The outliers (the outliers are mismatches, the interior points are matching points) of the image should be removed and the interior points must be retained based on this optimization result. For image matching, it is very effective to use RANSAC algorithm to remove mismatches.Mismatches should be removed after finding the 2D-3D matching points by RANSAC algorithm. The results of removing the mismatches are shown in Tab 1. The experimental results are shown in Fig.6.

TABLE I. REMOVAL OF MISMATCHES IN RANSAC OF CURRENT FRAME IMAGE

Image No.	In1	In2	In3	In4
Matching points	190	95	20	14
Mismatching points	74	40	19	13
the rest points	116	55	1	1

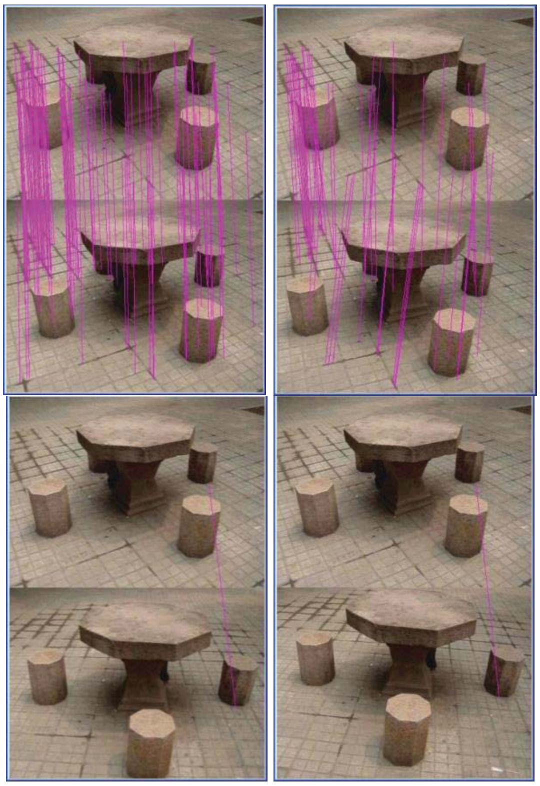

Fig. 6. Remove mismatches via RANSAC algorithm

Since the perspective of the current frame image is the closest one to that of the first image among the 4 reconstructed original images, and the perspective difference between the current frame image and the last two images is the largest, the number of matching points with the first image is the largest, while only one matching point can be found with the third and the fourth image after removing the mismatches by RANSAC algorithm. After the mismatches removed by RANSAC algorithm, there are 42 2D-3D matching points matched the current frame image and the reconstructed model. These matching points will be used as the input data for the following camera pose estimation.

7) Camera pose estimation

After searching by BBF algorithm and removing the mismatches by RANSAC algorithm, 42 2D-3D corresponding points are used as the input data of camera pose estimation. LM iterative optimization algorithm is used to optimize the reprojection error function of the image and obtain the camera pose matrix of the current frame image. The initial value of the iterative algorithm is the P matrix calculated by the plane linear algorithm. It takes 0.2 seconds for LM optimization of current frame image, and the reprojection error is 0.548 pixels.

The final experimental results of real-time tracking algorithm based on reconstruction model in augmented reality are shown in Fig.7. Six images are tracked via this algorithm, and the operation time of the algorithm is shown in Tab.2.

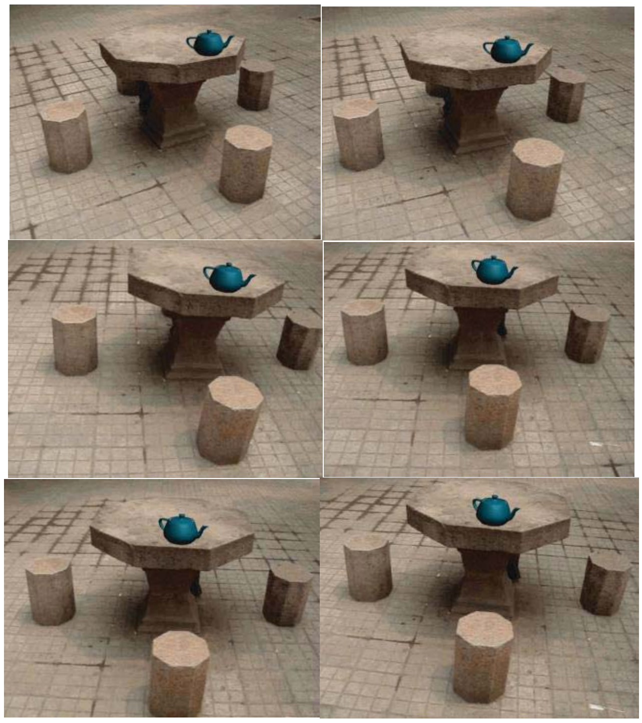

Fig. 7. Final Experimental Results

TABLE II. OPERATION TIME OF THE ALGORITHM (UNIT: SECOND)

Image No.	SIFT points extract	Feature match	Camera pose estimation	Total
1	5.059	0.736	0.15	5.945
2	5.863	0.601	0.28	6.744
3	6.049	0.676	0.16	6.885
4	6.145	0.719	0.20	7.064
5	6.129	0.663	0.26	7.052
6	5.34	0.672	0.18	6.192

As shown in Tab.2, single image tracking via this algorithm will cost about 6-7 seconds totally on the experimental environment described in the section above. And due to the low speed of SIFT points extraction, it takes quite a long time to track via the real-time algorithm in the experimental system. In order to improve the speed of sift extraction, the algorithm should be improved using GPU-SIFT extraction method, instead of the current method. Additionally it is also an effective method to improve the performance of the system hardware. The improved GPU-SIFT algorithm is used to extract SIFT points in a higher performance computer. The speed of extraction can be reduced to about 500ms, which can greatly improve the operation speed of the algorithm. Besides, in current experimental system, the reprojection error is about 0.5 pixels, which is almost no error in the visual range of human vision. Considering the operation environment constraint, long time consume does not affect the usage of this algorithm as long as there's no negative effect on the human vision.

B. Discussion and Improvement

Real-time visual tracking algorithm is successfully realized, that is, the system framework of real-time tracking algorithm based on reconstructed model and its performance is studied deeply. Compared with existing tracking algorithm, this algorithm can significantly improve the performance of a visual tracking system due to its high robustness and high accuracy.

In order to match 2D-3D points quickly and accurately, a new feature matching algorithm based on the combination of reconstructed model and k-d tree is applied. The feature matching algorithm based on reconstructed model is more robust and stable than traditional feature matching algorithm based on geometric feature and is able to solve occlusion issue, illumination issue and viewing angle change issue. The BBF algorithm based on k-d tree is able to find matching points in high-dimensional feature space quickly and accurately. According to the advantages of these two algorithms, the new feature matching algorithm based on the combination of reconstructed model and BBF algorithm is improved and beneficial.

As to the camera pose estimation based on LM iterative algorithm, camera P matrix of current frame is the core of the tracking algorithm. Considering re-projection error calculated from 2D-3D matching points exists, it is necessary to reduce this re-projection error as much as possible. The algorithm based on LM iterative can effectively and rapidly converge to minimum re-projection error, improving the performance of the real-time tracking system. Meanwhile, in order to improve system robustness, RANSAC algorithm is applied in the reconstructed model based real-time tracking algorithm for outliers filtering.

In the experiments, shortcomings of this algorithm still exist and should be focused on as follow.

a. Some errors are inevitable resulting from numerical calculation applied in the algorithm, leading to jitter issue. Several optimized algorithms can be taken into account, for instance, decreasing the number of mismatching points during 2D-3D matching by lowering the threshold of BBF algorithm; modifying objective function of LM algorithm; or adding filter to ensure the accuracy of P matrix in attitude estimation.

b. The implementation time of the algorithm depends on the sum time of SIFT point extraction, 2D-3D matching and attitude estimation. Currently, 2D-3D matching and LM iterative algorithm run quickly enough, however, SIFT point extraction runs slowly, which significantly impact the real-time performance of algorithm. GPU-based SIFT extraction algorithm can theoretically accelerate the calculation speed up to 20 times.

c. It is hardly to eliminate mismatching points in 2D-3D matching. Decreasing the number of mismatching points is the only way to avoid such mismatching as much as possible. One possible method is to eliminate mismatching points as much as possible to ensure every input point has positive impact on final result.

IV. Summary

Teleoperation mobile robots can give full play to the advantages of man and machine separately. The mode of teleoperation is an important feature of deep space exploration in the future. Based on the requirements of teleoperation tasks such as augmented reality teleoperation in unknown celestial environment, a new framework of real-time tracking algorithm was proposed and different algorithms including SIFT, BBF, LM etc., were analyzed and discussed. As a result, the steps of real-time tracking algorithm based on reconstruction model were established and several conclusions were drawn as below.

The operation speed of the algorithm is mainly determined by SIFT point extraction, 2D-3D feature matching and LM pose estimation. The experimental results show that speed of SIFT points extraction is relatively slow, while the one of 2D-3D feature matching and LM pose estimation are relatively fast.

The experimental system can establish the 2D-3D matching quickly and accurately, which ensure the accuracy and stability of the tracking results, due to the improvements on feature points matching of the algorithm, such as 2D-3D matching via BBF algorithm and removing mismatches by RANSAC algorithm. Additionally, the reprojection error is less than one pixel, that is, the accuracy of target tracking is in sub-pixel level. Therefore the algorithm has almost no error in visual range of human vision.

In the following research, SIFT algorithm needs to be improved so that the speed of extracting can be lifted. The visual tracking performance of augmented reality on orbit teleoperation system should be optimized for the further promotion, of the human-computer interaction experience in the process of on orbit teleoperation.

References

[1] Zeng Lingbin, Qiu Baogui, Xiao Jie. Research on the Roadmap and Typical Scenarios ofLunar Surface Robot Exploration[J]. 2015(21):263-269. (in Chinese)

[2] Zhu Miaoliang, Yao yuan, Jiang Yunliang. A review of augmented reality. Chinese Journal of image graphics, 2004.9 (07): PP. 767-774.

[3] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, Blair MacIntyre. Recent Advances in Augmented Reality. IEEE Computer Graphics and Applications, 2001(11-12): pp. 34-47.

[4] Lin Yuan. Design and implementation of teleoperation mode based on Augmented Reality Technology [D]. Harbin: Harbin University of technology, 2016: 1-56.

[5] Zhu Guangchao, Wang Tianmiao, Chou wusheng, et al. Research on Robot Teleoperation System Based on augmented reality [J]. Journal of system simulation, 2004, 5:943-946.

[6] R. Szeliski, S. Kang. Recovering 3D shape and motion from image streams using non-linear least squares. Technical report, Cambridge Research Laboratory, 1993.

[7] Iryna Skrypnyk, David Lowe. Scene Modelling, Recognition and Tracking with Invariant Image Features. IEEE, 2004

[8] R. Szeliski, S. Kang. Recovering 3D shape and motion from image streams using non-linear least squares. Technical report, Cambridge Research Laboratory, 1993