






3D Environment Modelling of Mobile Robot based on Virtual Reality and Point Cloud Technology

Qin Wang^{1*}, Anya Xiong² and Huirong Zhu³

^{1,2}Chongqing University of Posts and Telecommunications, Nan An, Chongqing 400065, China

³ChongQing Jiao Tong University, Nan An, Chongqing,400074, China

wangqin@cqupt.edu.cn, xiongay@cqupt.edu.cn, 3990060004003@cqjtu.edu.cn

Corresponding author: Wang Qin, wangqin@cqupt.edu.cn

Abstract. Traditional virtual reality (VR) technology generates indoor 3D map models through human modeling, which has problems such as slow response time and bias. For this reason, we propose a robot-aware environment modeling scheme with altered point cloud generation algorithm and VR. First, the 3D point cloud is reconstructed into a robot-sensitive environment model by environment-aware reconstruction algorithm and imported into the computational architecture. Then, the robot is repositioned in different environments by visual localization techniques and the robot pose is mapped in real time to achieve human-robot interaction. Experiments show that using visual repositioning techniques with point clouds to build environment-aware models is not only fast, but also solves the scene scale deviation and enables the reuse of maps. At the same time, the VR technology enables the operator to obtain a strong sense of immersion.

Keywords: visual simultaneous localization and map building; virtual reality; 3D modeling; indoor modeling

DOI: <https://doi.org/10.14733/cadaps.2023.S14.216-230>

1 INTRODUCTION

In recent years, with the development of science and technology, human exploration of unknown fields has become more frequent, such as space robot exploration, deep-sea exploration, nuclear robot detonation, and medical robotics in continuous development [8]. The scientific experiments conducted by humans under these fields urgently need fully autonomous robots to complete tasks instead of humans, such as performing operational tasks in some long-distance or dangerous environments. Although humans have made great breakthroughs in the field of intelligent robot development, and some intelligent robots that can assist humans in experimental tasks have been manufactured [5]. However, fully autonomous robots that are independent of human control have not yet been developed, and it will be difficult to make a breakthrough in the near future due to the limitations of sensing technology, mechanism design, control principles, artificial intelligence, and other technologies for fully autonomous intelligent robots [12],[7]. Therefore, the semi-autonomous

human-robot interaction teleoperation technology under the need of human participation is still the main means to achieve the detection of unknown important areas. Humans can operate semi-autonomous robots (e.g., robotic arms) at a distance to accomplish complex and dangerous tasks, among which space robotics for outer space exploration is an important application area for teleoperation. With teleoperated robotics, the operator can remotely control the robot to perform different tasks, and a typical teleoperated system structure is shown in Figure 1.

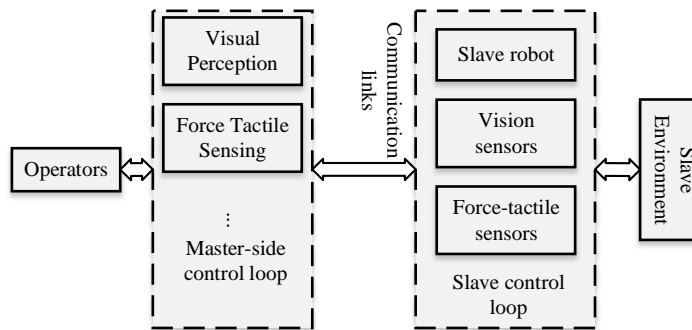


Figure 1: Typical remote operating system structure.

VR [3] simulates virtual environments by computer, thus giving a sense of immersion, and plays a great role in coal mines, security and other high-risk areas. However, its modeling efficiency is low and the requirements for modeling engineers are high, especially for real-time rendering of VR scenes modeling requirements are higher. Relatively speaking, 3D laser scanning modeling technology [13],[11] has many advantages such as fast, high accuracy, low cost, and no contact, but the amount of data is particularly huge and difficult to run in VR systems, and the scanning modeling of scenes generally needs to be used in conjunction with mapping equipment such as total stations, which requires high operator requirements

A more efficient way to construct a virtual 3D scene map of the real environment can be achieved by SLAM algorithm. The two mainstream methods of SLAM nowadays are LIDAR-based techniques [2],[22] and vision sensor-based techniques [15],[1]. Among them, LiDAR is not affected by lighting, and the study [8] used LOAM algorithm to construct point cloud maps in real time as 3D maps in virtual reality scenes, where each point color indicates its height, but the original color and texture information of the map is missing, and the point cloud map representation also makes the structure details blurred or missing. Visual SLAM can solve the problem of lack of texture in laser SLAM because the visual sensor has the advantages of small size, easy use, and rich texture and color information is stored in the image. The framework of current visual SLAM systems is relatively mature, among which ORB-SLAM2 [10] is the first open-source SLAM system for monocular, binocular and RGB-D with closed-loop, repositioning and map reuse modules, but only one camera sensor is used, which leads to poor system robustness. In contrast, VINS-Mono [16], a real-time SLAM framework for monocular visual inertial systems [25], is an optimization-based VIO approach with tight coupling of visual and IMU information, combined with closed-loop detection and map optimization, to construct a complete monocular visual SLAM approach. This method effectively solves the problem of scale uncertainty on monocular, but only sparse 3D maps are constructed. The study [20] constructs a dense 3D map based on a large-scale 3D reconstruction method with symbolic distance function, using sub-maps to reduce the cost of storage and optimization, and the system assumes that the gravity direction can be obtained in real time to reduce the dimensionality of the positional estimation problem.

To address the above problems of inefficiency and lack of textural structure information in real environment 3D modeling, this paper combines visual SLAM technology and VR technology to propose a VR-based real environment 3D modeling system for mobile robots. The indoor scenes are image captured, processed to generate dense 3D point cloud maps, and then the point cloud maps are reconstructed by surface reconstruction algorithm to generate 3D models and mapping, and imported into the virtual reality device for visualization to achieve an all-around view of the indoor 3D scenes and provide users with effective 3D information more quickly. After experimental verification, the proposed system effectively solves the problems of low efficiency of previous manual modeling and lack of texture and structure information in laser SLAM modeling. Finally, the effectiveness of the repositioning module is verified, and the reusability of the map model is realized when there is no significant change in the real environment, thus making the human-machine interaction more efficient.

2 VIRTUAL ROBOTIC ARM GEOMETRY MODELING

For the 3D virtual prediction environment built in this system, its concrete implementation is displayed on the picture control on MFC using the OpenGL graphics programming library on the Visual studio 2010 development platform [6]. There are two main methods for rendering 3D scenes based on OpenGL, one is to directly use the library functions in OpenGL to draw the basic units such as points and lines and 3D bodies, and build the required model by translation, rotation, scaling and other library functions, this method is simple and suitable for the construction of simple 3D models, but for complex model rendering can not be well restored, and the fidelity effect is poor [4]. The other is to use 3D graphics drawing software to draw the 3D model, and render the whole connected model by importing the model surface with fixed points [21]. In this paper, the second approach is to build a virtual environment based on SolidWorks modeling and OpenGL programming. The main implementation is to build a realistic 3D virtual model through coordinate transformation by importing the STL file generated by SolidWorks and calling the geometric function mathematical library such as GLM (OpenGL Mathematics) by OpenGL. The geometry modeling of the virtual arm is shown in Figure 2.

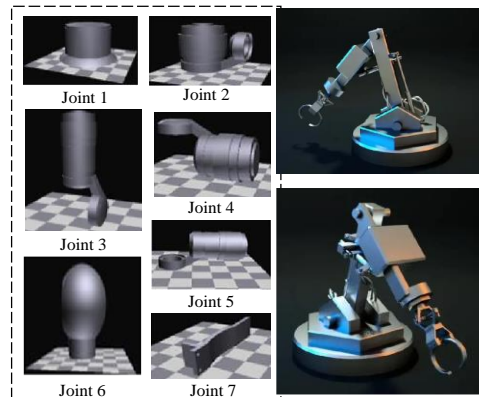


Figure 2: Geometric modeling of the virtual robot arm.

The virtual robot arm constructed by geometric transformation of the OpenGL model library has the same scale dimensions as the real SCHUNK robot arm. The left side shows the independent 3D model of each joint module, and the kinematic model is established by the kinematic principle of the

robot arm, and the robot inverse operation can be performed according to the kinematic model of the robot arm. The constructed virtual robot arm is shown on the right side of Figure 2, and the vertical state and initialization state of the robot arm are shown respectively.

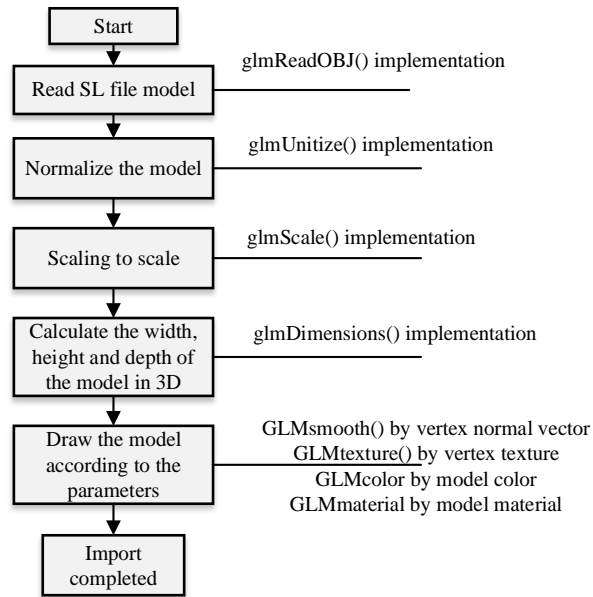


Figure 3: Flowchart of concrete implementation.

In order to make the constructed virtual 3D environment model more realistic, it is necessary to set the light source properties for the whole 3D model. The `glmLightfv(1,2,3)` function is used to set the light source, where the first parameter represents the label of the light source, the second parameter represents the ambient light, diffuse reflection, position and other property settings, and the third parameter represents a vector to set specific parameters to enhance the 3D rendering effect.

3 SYSTEM FRAMEWORK

The VR-based fast 3D modeling system for realistic environments of mobile robots consists of two modules, the Robot side and the Virtual Reality side, and the system framework is shown in Figure 1. As shown in the Robot side in Figure 1, the robot acquires the corresponding data using RGBD camera and IMU, estimates the robot pose through the front-end of visual odometry, and the Voxgraph algorithm is used as the back-end to optimize the pose and construct the global map, and the map is represented in the form of 3D point cloud, and the 3D point cloud map is reconstructed by the surface to generate the 3D model map, and then sent to the VR side. As shown in Figure 4, the operator makes a decision based on the real-time video stream and the robot pose in the 3D model map, plans the path in time and issues commands through the VR handle to control the robot movement, while the robot pose can be updated and mapped in the 3D model map in real time by repositioning.

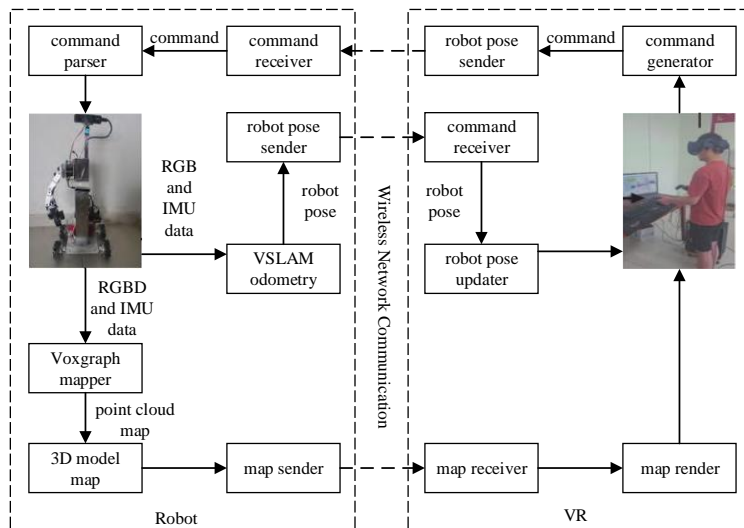


Figure 4: Framework of the proposed system.

3.1 Point Cloud Map and Repositioning

The robot is controlled by the VR handle to complete the data acquisition, and then the dense 3D point cloud scene map is generated by the visual SLAM algorithm. The block diagram of the visual SLAM algorithm is shown in Figure 5. Voxgraph is used as the main framework of the algorithm, and the whole algorithm consists of two modules: front-end and back-end. In the front-end, VINS-Mono is used as the vision-inertial guidance odometer to generate the sub-point cloud map by initially estimating the positional attitude, and to complete the computation of the three constraints required in the back-end. In the back-end, least squares are constructed using three constraints (alignment constraints, odometer constraints, and loopback constraints) to complete the optimization of the positional map.

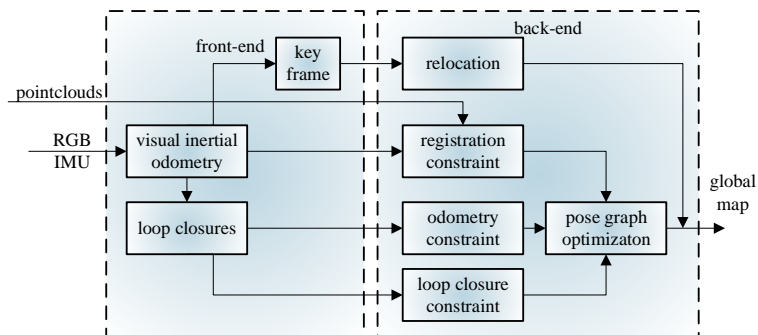


Figure 5: Visual SLAM system block diagram.

$$\begin{aligned}
& \arg_x \min \sum_{(i,j) \in R} \| e_{reg}^{i,j}(T_{WS_i}, T_{WS_j}) \|_{\sigma_r} + \\
& \sum_{(i,j) \in O} \| e_{odom}^{i,j}(T_{WS_i}, T_{WS_j}) \|_{\sum_o}^2 + \\
& \sum_{(i,j) \in L} \| e_{loop}^{i,j}(T_{WS_i}, T_{WS_j}) \|_{\sum_L}^2
\end{aligned} \tag{1}$$

Where: $x = \{T_{WS1}, T_{WS2}, \dots, T_{WSN}\}$ is the initial estimated poses of the odometer, $T_{WSi} \in R^4$; R, O, L are the index sets of poses in the alignment constraint, odometer constraint, and loopback constraint, respectively. Based on this, a repositioning module is added to align the current sliding window frame with the past pose-map when the robot enters the environment again, to judge whether the similarity between the feature points in the current image and the feature points in the pose-map is greater than the set threshold, and if the threshold is exceeded, the successful repositioning is judged, and the repositioning is realized by inverse depth solving the camera's poses in the map. The successful repositioned poses are transmitted to the VR side, and the corresponding spatial coordinates of the robot can be displayed in the model.

3.2 Generating the Model Map

In order to realize the 3D modeling of the real scene, the dense 3D point cloud needs to be processed by the surface reconstruction algorithm to obtain the refined 3D scene model and the corresponding texture map. The surface reconstruction algorithm in this paper is based on Possion reconstruction [14], and the basic steps are as follows.

- 1) First, the point cloud map is down-sampled by statistical filter for noise removal [19] and the farthest point sampling (FPS) method [22], and the existence of outlier points will cause the surface of the reconstructed object to be unsmooth, so the statistical filter is used to remove the outlier points. For any point p_i , the neighborhood is obtained by $KnnSearch$ $p_j \subset Nhd(p_i)$, and the average

Euclidean distance x from p_i to the points in its neighborhood is calculated, and let all the average distances satisfy a Gaussian distribution as follows.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{2}$$

Where: μ is the mean value; σ is the standard deviation. The points whose mean distance is outside the range of standard deviation are eliminated, i.e., the outliers are eliminated. Then the point cloud map is down-sampled by FPS. The point P_0 is randomly selected as the starting point in

the set A with the input point cloud number N, and the sampling set $B = \{p_0\}$ is obtained. Next, the distance from the remaining points in the set A to the points in the set B is calculated, and the

point with the farthest distance is selected as p_1 and added to the set B. And when the number of points in the set B is greater than 1, the sampling principle satisfies.

$$\max \left(\min \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,N'} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,N'} \\ \vdots & \vdots & & \vdots \\ d_{N-2,1} & d_{N-2,2} & \cdots & d_{N-2,N'} \\ d_{N-1,1} & d_{N-1,2} & \cdots & d_{N-1,N'} \end{bmatrix} \right) \quad (3)$$

Where: d_i, j denotes the Euclidean distance from the remaining points in set A to the sampled points in set B. The minimum value of each column is first taken as the distance from the point in set A to set B, and then the point with the maximum distance in that row is taken as the sampling point, and the sampling step is repeated until N' target sampling points are sampled. The purpose of down-sampling points is to improve the speed of reconstruction on the basis of satisfying the complete requirements of model reconstruction, and the farthest point sampling can cover all the points in the space as much as possible

- 2) Calculate the normal of each point. For any point p_i , the neighborhood $p_j \in \text{Nbhd}(p_i)$ is obtained by KnnSearch, and the covariance matrix is constructed by calculating the center of mass

$$o_i = \frac{1}{k} \sum_{j=0}^{j=k} p_j \quad \text{according to the neighborhood.}$$

$$\mathbf{R}_{\text{cov}} = \frac{1}{k} \sum_{p_j \in \text{Nbhd}(p_i)} (\mathbf{p}_j - \mathbf{o}_i) \cdot (\mathbf{p}_j - \mathbf{o}_i)^T \quad (4)$$

- 3) Generate a model using Possion surface reconstruction. the core idea of Possion surface reconstruction [23] is that given a point cloud region S, the point cloud represents the location of the object surface M, its normal vector n represents the inner and outer directions, and the point cloud boundary is ∂_M , by implicitly fitting an indicator function χ_M derived from the object.

$$\chi_M(q) = \begin{cases} 1, q \in M \\ 0, q \notin M \end{cases} \quad (5)$$

Where: q denotes the position of a point in the point cloud region. This transforms the problem of reconstructing a point cloud surface $M = \partial_M$ into a problem of solving the reconstruction indicator function χ_M . The basic principle of the Possion surface reconstruction algorithm is to create an octree \mathcal{G} topology of depth D for the input point set and to attach a nodal function F_o to each node o . The expansion is of the form

$$F_o(\mathbf{q}) \equiv F\left(\frac{\mathbf{q}-\mathbf{c}_o}{w_o}\right) \times \frac{1}{w_o^3} \quad (6)$$

Where: \mathbf{c}_o denotes the center of node o ; w_o denotes the width of node o . Introduce the basis function F to describe the relationship between the node function and the vector field \mathbf{V} .

$$F(\mathbf{q}) = F\left(\frac{\mathbf{q}}{2^D}\right) \quad (7)$$

The introduction of a high-pass filter converts the surface reconstruction of a directed point set into a spatial Poisson problem.

$$F(x, y, z) = [B(x)B(y)B(z)]^* \quad (8)$$

$$B(t) = \begin{cases} 1, & |t| < 0.5 \\ 0, & |t| \geq 0.5 \end{cases} \quad (9)$$

Denote the 8 neighboring nodes of the current node p_s by $\text{Nbr}_p(s)$ and use $\{\alpha_{o,s}\}$ as the interpolated weights so that the vector field of the surface gradient domain represented by the indicator function is approximated as

$$\mathbf{V}(\mathbf{q}) \equiv \sum_{s \in S} \sum_{o \in \text{Nbr}_p(s)} \alpha_{o,s} F_o(\mathbf{q}) \mathbf{n}_o \quad (10)$$

The problem of solving Poisson's equation $\Delta \chi = \nabla \cdot \mathbf{V}$ is simplified as

$$\sum_{o \in \mathcal{G}} \|\langle \Delta \chi - \nabla \cdot \mathbf{V}, \mathbf{F}_o \rangle\|^2 = \sum_{o \in \mathcal{G}} \|\langle \Delta \chi, \mathbf{F}_o \rangle - \langle \nabla \mathbf{V}, \mathbf{F}_o \rangle\|^2 \quad (11)$$

For a given $|\mathcal{G}|$ -dimensional vector \mathbf{V} , the solution to the function χ can be solved by approximating \mathbf{V} with a vector composed of the Laplace operator and \mathbf{F}_o projected by this

function in the function space, i.e., by converting it into a solution to $\sum_{o \in \mathcal{G}} \|\mathbf{v} - \mathbf{v}_o\|^2$.

To solve the vector using matrix operations, the algorithm defines a $\nabla \cdot \mathbf{V}$ -order matrix L . Each item of the matrix is the result of the dot product of the Laplace operator and each nodal function, and the

problem of solving for χ is finally reduced to solving for $\min_{x \in \mathbf{R}^{(v)}} \|\mathbf{Lx} - \mathbf{v}\|^2$. Finally, the equivalence surface is extracted by the moving cube method to obtain the reconstructed model.

- 4) Point cloud triangulation to generate texture mapping. The points on the surface S are triangulated, and the vertices along the triangulation are mapped to the mapping plane UV coordinates, and the mapping color information is the average of the grayscale of the three channels corresponding to the three points.

$$I_i = \frac{I_{i1} + I_{i2} + I_{i3}}{3} \quad (12)$$

Where: i is taken as r , g , and b , representing the grayscale values of R, G, and B channels, respectively.

- 5) Match the model and texture map. According to the normal direction of the point in step 2), the texture map is matched with the model side of the normal positive direction to complete the construction of the whole map model.

4 EXPERIMENT AND DEMONSTRATION

4.1 Point Cloud Generation

The repositioning module in the system is evaluated on the basis of real scene modeling, and the initial positioning time, repositioning time, and repositioning accuracy of the system are tested several times and averaged. As shown in Figure 6, the real-time positioning coordinates will be mapped in the 3D model; the real-time video stream will assist the user to know the surroundings of the robot location; the viewpoint can be switched to the robot to observe the virtual environment in which the robot is located.

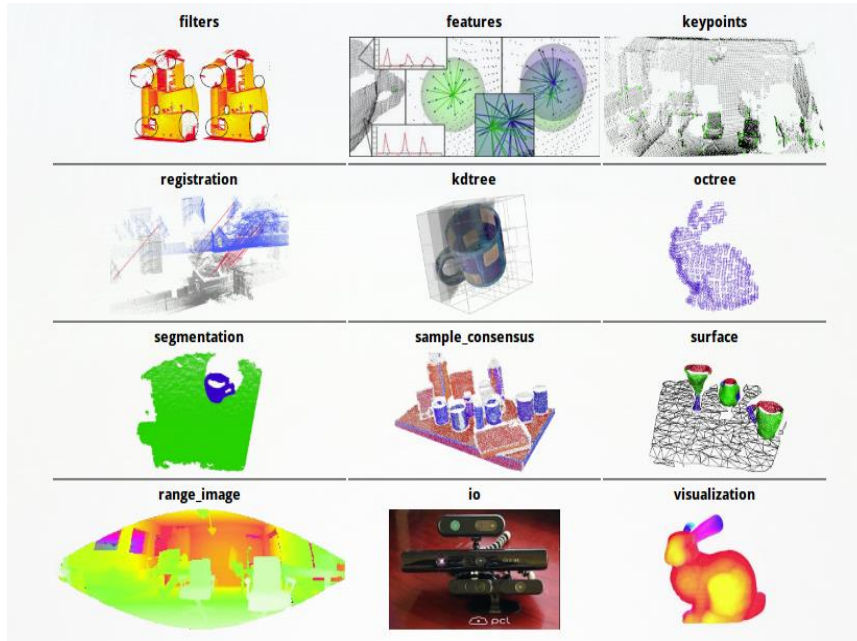


Figure 6: Real-time positioning verification.

Table 1 shows the initial positioning and repositioning time consumption statistics. In the 20 time-consuming tests, the system was successfully initialized and repositioned each time, and the average error of repositioning was 0.31 m, which proved that the repositioning mode was not effective. The effectiveness and stability of the repositioning module were verified.

<i>Parameter</i>	<i>Value</i>
<i>Initial positioning time / s</i>	<i>11. 853</i>
<i>Relocation time / s</i>	<i>17. 586</i>
<i>Relocation accuracy / m</i>	<i>0. 32</i>

Table 1: Initial positioning and repositioning time-consuming tests.

A mobile robot Pioneer-3DX was used to track a moving target in a laboratory with an area of 80m². The maximum speed of the robot is set to 200m/s and the maximum angular speed is 10(°)/s, and the camera is mounted above the robot to acquire images every 3s.

4.2 Performance Analysis

Table 2 shows the record table of the number of point clouds after applying the method of this paper with the motion target tracking method based on the improved Camshift algorithm of literature [19] and the improved motion target tracking method based on ECO-HC of literature [9] to segment the images with the number of point clouds of 30700 at the first frame, 200 frames, 400 frames and 600 frames.

<i>Number of frames</i>	<i>Method of this paper</i>	<i>Literature [19] method</i>	<i>Literature [9] method</i>
<i>1</i>	<i>195128</i>	<i>166551</i>	<i>152891</i>
<i>200</i>	<i>121862</i>	<i>112006</i>	<i>153805</i>
<i>400</i>	<i>135897</i>	<i>125179</i>	<i>155120</i>
<i>600</i>	<i>194002</i>	<i>174053</i>	<i>155869</i>

Table 2: Record table of the number of point clouds after segmentation by the three methods.

Using the sample data after segmentation by the three methods shown in Table 2, the root-mean-square error of point cloud segmentation by the three methods with different segmentation ratios was counted, and the results are shown in Figure 7.

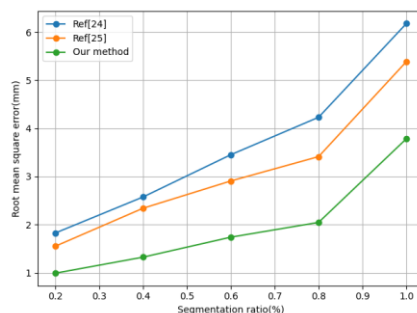


Figure 7: Root-mean-square error curve of point cloud segmentation.

From Figure 7, it can be seen that, under the same segmentation ratio, the RMS error variation curve of point cloud segmentation by applying this paper is always lower than that of the method of literature [19] and the method of literature [9], and the curve is relatively flat. It indicates that the error of point cloud segmentation by applying the method in this paper is smaller and more suitable for practical needs under the same segmentation ratio.

Figure 8 represents the distance accuracy of the moving target at four angular velocities obtained by applying the method of this paper, and this index is reflected by the change of distance coefficient R_i at four angular velocities, and the larger the distance coefficient R_i is, the lower the tracking accuracy is. From the analysis of Figure 8, it can be seen that the distance coefficient R_i is the smallest when the angular velocity value is $6(^{\circ})/s$ and $8(^{\circ})/s$, and the distance accuracy of motion target tracking obtained by applying this paper is higher than the distance accuracy of motion target tracking when the angular velocity is $4(^{\circ})/s$ and $10(^{\circ})/s$, and the highest value is reached when the center distance deviation is 9 pixels. This indicates that the mobile robot has better motion target tracking when using angular velocity $6(^{\circ})/s$ and $8(^{\circ})/s$ for motion target tracking.

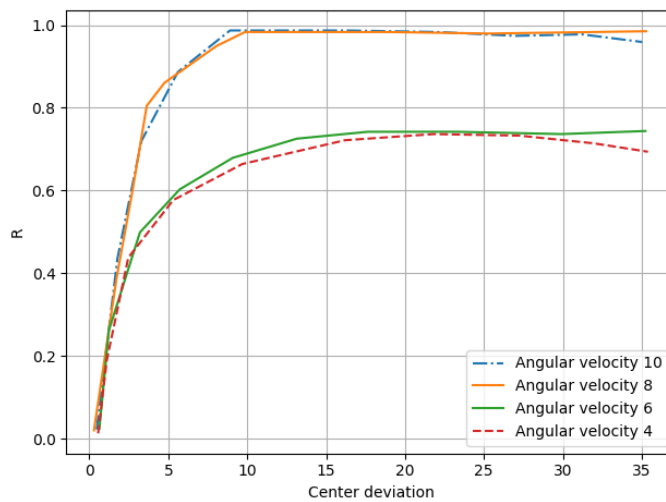


Figure 8: Distance accuracy curve of moving target.

Figure 9 shows the center distance error at four angular velocities obtained by applying the method of this paper. From Figure 9, it can be seen that the center distance error of the motion target tracking obtained by applying the method of this paper is lower than the center distance error of the motion target tracking at the angular velocity values of $6(^{\circ})/s$ and $8(^{\circ})/s$ than that at the angular velocities of $4(^{\circ})/s$ and $10(^{\circ})/s$, and there is no large fluctuation in the center distance error curve and the target tracking is stable. Figure 10 represents the target tracking area of the robot in the absence of obstacles versus in the presence of obstacles. Figure 11 represents the error offsets of the target tracking region obtained by applying the method of this paper with the method of literature [19] for 240 experiments on the target tracking region in Figure 11. From Figure 11, it can be seen that the target tracking offset curve obtained by applying the method in this paper is always lower than the method in literature [9] under the same number of experiments, no matter in the case of obstacles or no obstacles, and the target tracking offset curve obtained by applying the method in this paper changes more smoothly without large fluctuations, while the target tracking offset curve of the method in literature [9] and the method in literature [19] fluctuates more. It means that the tracking stability of the target is stronger and the error is smaller by applying the method in this paper.

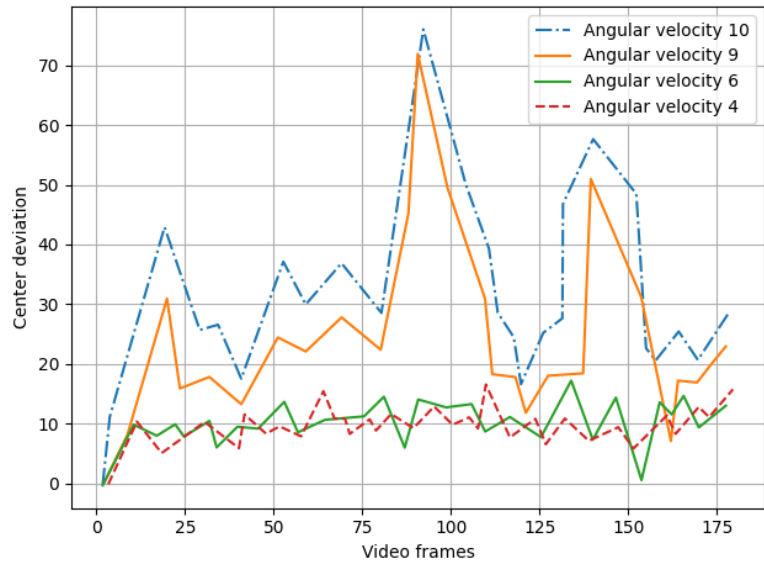


Figure 9: Center distance error.

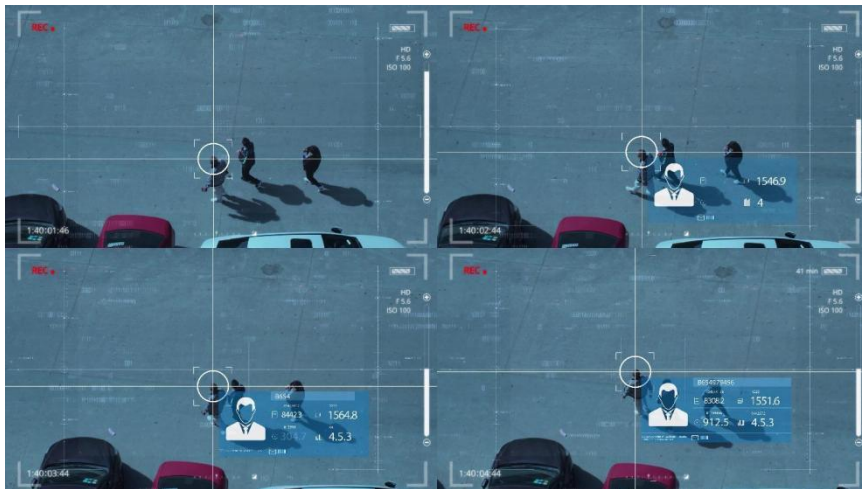
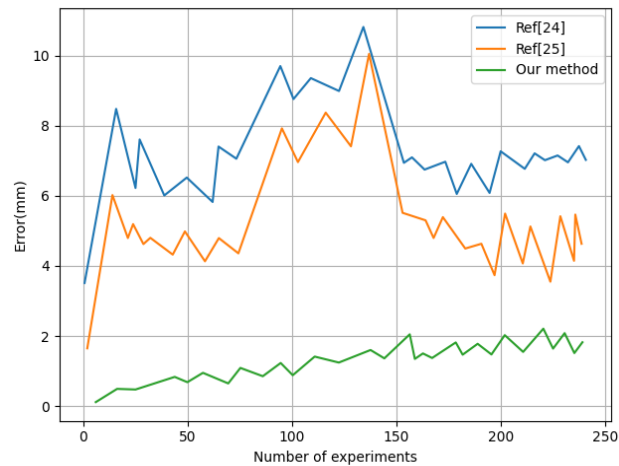
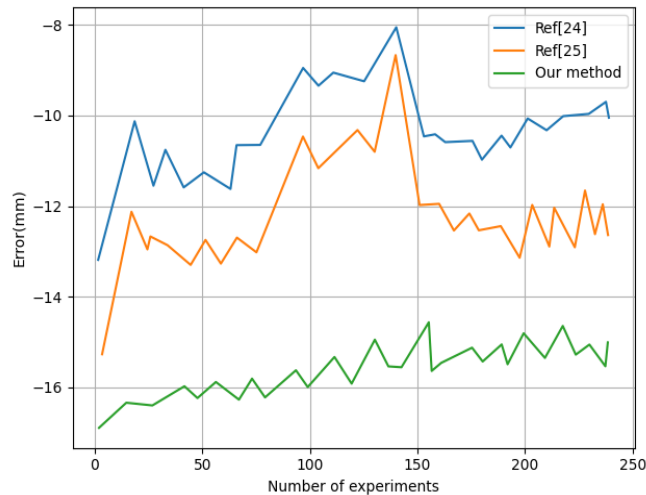


Figure 10: Schematic diagram of the target tracking region.



(a) No Obstacles



(b) Obstacles

Figure 11: Target tracking error offset curve.

5 CONCLUSION

In this paper, the motion target region is determined based on point cloud segmentation, and the tracking of the motion target by the mobile robot is achieved by using the hybrid servo tracking method in the process of determining the region. And the following conclusions are drawn. The feasibility and effectiveness of the method to accomplish the tracking task are verified, and the distance accuracy of its motion target tracking is higher than other motion target tracking methods, and the highest value of motion target tracking distance accuracy is achieved at the center distance deviation of 9 pixels; and the center distance error in motion target tracking can be reduced by using the appropriate angular speed. However, it is not possible to verify whether this method is also

better than other motion target tracking methods in multi-target tracking because of the single target tracked by the mobile robot, and the next step will be to study the direction of multi-target tracking by the mobile robot.

Qin Wang, <https://orcid.org/0000-0003-1481-7954>

Anyu Xiong, <https://orcid.org/0000-0002-7502-0247>

Huirong Zhu, <https://orcid.org/0009-0005-7200-5454>

Funding Statement

This study is supported by Chongqing Municipal Education Commission, Humanities and Social Sciences Research Project, Project name: Research on the construction of "Red Discourse"-based on virtual reality Internet, Project Number: 22SKJD080.

REFERENCES

- [1] Atiyah, H. A.; Hassan, M. Y.: Outdoor Localization in Mobile Robot with 3D LiDAR Based on Principal Component Analysis and K-Nearest Neighbors Algorithm, *Engineering and Technology Journal*, 39(06), 2021, 965-976. <https://doi.org/10.30684/etj.v39i6.2032>
- [2] Bavelos, A. C.; Kousi, N.; Gkournelos, C.; Lotsaris, K.; Aivaliotis, S.; Michalos, G.; Makris, S.: Enabling flexibility in manufacturing by integrating shopfloor and process perception for mobile robot workers, *Applied Sciences*, 11(9), 2021, 3985. <https://doi.org/10.3390/app11093985>
- [3] Cui, X.; Lu, C.; Wang, J.: 3D semantic map construction using improved ORB-SLAM2 for mobile robot in edge computing environment, *IEEE Access*, 8, 2020, 67179-67191. <https://doi.org/10.1109/ACCESS.2020.2983488>
- [4] Han, L.; Zheng, T.; Zhu, Y.; Xu, L.; Fang, L.: Live semantic 3d perception for immersive augmented reality, *IEEE transactions on visualization and computer graphics*, 26(5), 2020, 2012-2022. <https://doi.org/10.1109/TVCG.2020.2973477>
- [5] Kim, P.; Park, J.; Cho, Y. K.; Kang, J.: UAV-assisted autonomous mobile robot navigation for as-is 3D data collection and registration in cluttered environments, *Automation in Construction*, 106, 2019, 102918. <https://doi.org/10.1016/j.autcon.2019.102918>
- [6] Kolhatkar, C.; Wagle, K.: Review of SLAM algorithms for indoor mobile robot with LIDAR and RGB-D camera technology, *Innovations in Electrical and Electronic Engineering: Proceedings of ICEEE 2020*, 2021, 397-409. https://doi.org/10.1007/978-981-15-4692-1_30
- [7] Li, X.; Du, S.; Li, G.; Li, H.: Integrate point-cloud segmentation with 3D LiDAR scan-matching for mobile robot localization and mapping, *Sensors*, 20(1), 2019, 237. <https://doi.org/10.3390/s20010237>
- [8] Linxi, G. O. N. G.; Yunfei, C. A. I.: Human Following for Outdoor Mobile Robots Based on Point-Cloud's Appearance Model, *Chinese Journal of Electronics*, 30(6), 2021, 1087-1095. <https://doi.org/10.1049/cje.2021.07.017>
- [9] Liu, H.; Song, R.; Zhang, X.; Liu, H.: Point cloud segmentation based on Euclidean clustering and multi-plane extraction in rugged field, *Measurement Science and Technology*, 32(9), 2021, 095106. <https://doi.org/10.1088/1361-6501/abead3>
- [10] Lv, Z.; Lloret, J.; Song, H.: Real-time image processing for augmented reality on mobile devices, *Journal of Real-Time Image Processing*, 18, 2021, 245-248. <https://doi.org/10.1007/s11554-021-01097-9>
- [11] Mac, T. T.; Lin, C. Y.; Huan, N. G.; Duc, L.; Nhat, P. C. H.; Hai, H. H.: Hybrid SLAM-based exploration of a mobile robot for 3D scenario reconstruction and autonomous navigation, *Acta Polytech. Hung*, 18, 2021, 197-212. <https://doi.org/10.12700/APH.18.6.2021.6.11>

- [12] Maset, E.; Scalera, L.; Beinat, A.; Visintini, D.; Gasparetto, A.: Performance investigation and repeatability assessment of a mobile robotic system for 3D mapping, *Robotics*, 11(3), 2022, 54. <https://doi.org/10.3390/robotics11030054>
- [13] Mu, L.; Yao, P.; Zheng, Y.; Chen, K.; Wang, F.; Qi, N.: Research on SLAM algorithm of mobile robot based on the fusion of 2D LiDAR and depth camera, *IEEE Access*, 8, 2020, 157628-157642. <https://doi.org/10.1109/ACCESS.2020.3019659>
- [14] Ostanin, M.; Yagfarov, R.; Klimchik, A.: Interactive robots control using mixed reality. *IFAC-PapersOnLine*, 52(13), 2019 ,695-700. <https://doi.org/10.1016/j.ifacol.2019.11.307>
- [15] Pan, Y.; Chen, C.; Li, D.; Zhao, Z.; Hong, J.: Augmented reality-based robot teleoperation system using RGB-D imaging and attitude teaching device, *Robotics and Computer-Integrated Manufacturing*, 71, 2021, 102167. <https://doi.org/10.1016/j.rcim.2021.102167>
- [16] Park, K. B.; Choi, S. H.; Kim, M.; Lee, J. Y.: Deep learning-based mobile augmented reality for task assistance using 3D spatial mapping and snapshot-based RGB-D data, *Computers & Industrial Engineering*, 146, 2020, 106585. <https://doi.org/10.1016/j.cie.2020.106585>
- [17] Pérez, L.; Rodríguez-Jiménez, S.; Rodríguez, N.; Usamentiaga, R.; García, D. F.: Digital twin and virtual reality-based methodology for multi-robot manufacturing cell commissioning, *Applied sciences*, 10(10), 2020, 3633. <https://doi.org/10.3390/app10103633>
- [18] Rosas-Cervantes, V.; &Lee, S. G.: 3D localization of a mobile robot by using Monte Carlo algorithm and 2D features of 3D point cloud, *International Journal of Control, Automation and Systems*, 18(11), 2020, 2955-2965. <https://doi.org/10.1007/s12555-019-0313-0>
- [19] Tian, Y.; Wang, X.; Shen, Y.; Guo, Z.; Wang, Z.; Wang, F. Y.: Parallel point clouds: Hybrid point cloud generation and 3D model enhancement via virtual-real integration, *Remote Sensing*, 13(15), 2021, 2868. <https://doi.org/10.3390/rs13152868>
- [20] Wang, C.; Wen, C.; Dai, Y.; Yu, S.; Liu, M.: Urban 3D modeling with mobile laser scanning: a review. *Virtual Reality & Intelligent Hardware*, 2(3), 2020, 175-212. <https://doi.org/10.1016/j.vrih.2020.05.003>
- [21] Wang, H.; Zhang, C.; Song, Y.; Pang, B.; Zhang, G.: Three-dimensional reconstruction based on visual SLAM of mobile robot in search and rescue disaster scenarios, *Robotica*, 38(2), 2020, 350-373. <https://doi.org/10.1017/S0263574719000675>
- [22] Wang, J.; Tao, B.; Gong, Z.; Yu, W.; Yin, Z.: A mobile robotic 3-D measurement method based on point clouds alignment for large-scale complex surfaces, *IEEE Transactions on Instrumentation and Measurement*, 70, 2021, 1-11. <https://doi.org/10.1109/TIM.2021.3090156>
- [23] Wang, X.; Mizukami, Y.; Tada, M.; Matsuno, F.: Navigation of a mobile robot in a dynamic environment using a point cloud map, *Artificial Life and Robotics*, 2021, 26, 10-20. <https://doi.org/10.1007/s10015-020-00617-3>
- [24] Xiang, S.; Wang, R.; Feng, C.: Mobile projective augmented reality for collaborative robots in construction, *Automation in Construction*, 2021, 127, 103704. <https://doi.org/10.1016/j.autcon.2021.103704>
- [25] Zhou, T.; Zhu, Q.; Du, J.: Intuitive robot teleoperation for civil engineering operations with virtual reality and deep learning scene reconstruction, *Advanced Engineering Informatics*, 46, 2020, 101170. <https://doi.org/10.1016/j.aei.2020.101170>