electronics

MDPI

Article

Research of Hand–Eye System with 3D Vision towards Flexible Assembly Application

Peidong Liang [1], Wenwei Lin [1], Guantai Luo [1] and Chentao Zhang [1,2,*]

[1] Fujian (Quanzhou)-HIT Research Institute of Engineering and Technology, Quanzhou 362000, China; lpd0004@hitqz.com (P.L.); lww2127@hitqz.com (W.L.); lgt2091@hitqz.com (G.L.)
[2] Department of Instrumental and Electrical Engineering, Xiamen University, Xiamen 361000, China
* Correspondence: zhangct@xmu.edu.cn; Tel.: +86-159-5928-5107

Abstract: In order to improve industrial production efficiency, a hand–eye system based on 3D vision is proposed and the proposed system is applied to the assembly task of workpieces. First, a hand–eye calibration optimization algorithm based on data filtering is proposed in this paper. This method ensures the accuracy required for hand–eye calibration by filtering out part of the improper data. Furthermore, the improved U-net is adopted for image segmentation and SAC-IA coarse registration ICP fine registration method is adopted for point cloud registration. This method ensures that the 6D pose estimation of the object is more accurate. Through the hand–eye calibration method based on data filtering, the average error of hand–eye calibration is reduced by 0.42 mm to 0.08 mm. Compared with other models, the improved U-net proposed in this paper has higher accuracy for depth image segmentation, and the A_{cc} coefficient and D_{ice} coefficient achieve 0.961 and 0.876, respectively. The average translation error, average rotation error and average time-consuming of the object recognition and pose estimation methods proposed in this paper are 1.19 mm, 1.27°, and 7.5 s, respectively. The experimental results show that the proposed system in this paper can complete high-precision assembly tasks.

Keywords: hand–eye calibration; U-net; point cloud registration

Citation: Liang, P.; Lin, W.; Luo, G.; Zhang, C. Research of Hand–Eye System with 3D Vision towards Flexible Assembly Application. *Electronics* **2022**, *11*, 354. https://doi.org/10.3390/electronics11030354

Academic Editor: Jeha Ryu

Received: 28 December 2021
Accepted: 19 January 2022
Published: 24 January 2022

1. Introduction

The field of automatic robotic assembly has attracted much attention. In recent years, automatic robotic assembly technology has been gradually applied to various fields such as automobiles, aerospace, and electronics manufacturing. The application of automatic robotic assembly technology has greatly improved the production efficiency of enterprises. In automatic robotic assembly tasks, the robot is guided by vision sensors or force/torque (F/T) sensors to complete the assembly work.

The automatic robotic assembly system based on force/torque sensors senses the force of the workpiece in the assembly process, and guides the robot to complete the assembly task by analyzing the force model and adjusting feedback system. Peng et al. [1] designed a novel three-layer pose adjustment mechanism consisting of two parallel mechanisms as a force sensor to assist robots in completing automatic assembly tasks. Wang et al. used an elastic displacement device to sense errors in the assembly process and assist the robot in automatic assembly through closed-loop feedback [2]. Zeng et al. proposed an external force/torque calculation algorithm based on dynamic model identification to realize the flexible assembly of robots [3]. Gai et al. proposed a compliance control method to solve the insertion assembly problem [4]. Park et al. proposed a compliant nail hole assembly method based on blind search using spiral force trajectory (SFT) [5].

Though the automatic assembly problem can be solved with the assistance of a force/torque sensor, it still does not appear to be "flexible." Therefore, researchers usually equip the system with vision sensors to assist the robot in more flexible automatic assembly.

Ma et al. built an assembly system consisting of a robot, three cameras, a micro-force sensor and a specific gripper [6]. Li et al. proposed a three-dimensional visual method for object pose estimation coupled with admittance control to promote robotic shaft-in-hole assembly [7]. Liu et al. solved the peg-in-hole precise assembly problem by combining microscopic vision and force information. [8]. Qin et al. proposed a precision assembly method based on multi-camera micro-vision and three-dimensional force feedback [9]. Song et al. proposed robotic assembly skill learning with deep Q-learning using visual perspectives and force sensing to learn an assembly policy [10]. Wang et al. developed a high-precision assembly system combining robotic vision servo technology and robot force feedback control technology [11].

While the combination of a force/torque sensor and a vision sensor can solve the problem of automatic assembly by a robot, the force/torque sensor is expensive, which will increase the cost of the entire system. In order to improve assembly efficiency and reduce system cost, researchers are committed to the research of automatic robotic assembly system using only a visual sensor as the auxiliary. For example, an automatic assembly system based on stereo vision was researched by Chang et al. [12], which completes the precise assembly of mobile phone cases. Jiang et al. proposed a calibration method for the large-scale cabin assembly system (LCSS) with visual guidance [13]. Dong et al. realized robot assembly pose estimation through point cloud registration [14]. Yan et al. used a structured light 3D camera to build a high-precision robot assembly system to achieve high-precision assembly of two workpieces [15]. Litvak et al. proposed a high-precision two-stage attitude estimation method based on deep learning to realize automated assembly of workpieces [16]. Li et al. proposed an automatic vision positioning for precise grasping of workpieces in the assembly process [17].

For robot flexible automatic assembly tasks, this paper designs a hand–eye system based on the 3D vision, which includes two modules: hand–eye calibration and automatic assembly. The working flow chart of the system designed in this paper is shown in Figure 1.

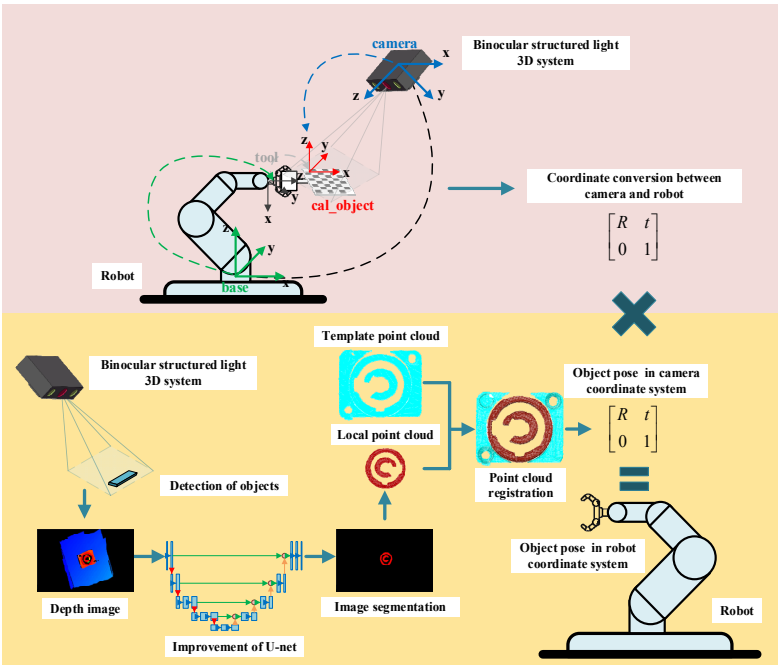


Figure 1. Workflow of automatic robotic assembly system. The transformation relationship between the robot and the camera is obtained through hand–eye calibration. Through point cloud segmentation and point cloud registration, the pose of the target in the camera coordinate system is calculated. The hand–eye transformation matrix is multiplied by the pose of the target in the camera coordinate system to obtain the pose of the target in the robot coordinate system, which realizes automatic assembly guidance of the robot.

The main contributions of this paper are as follows:

1. A hand–eye calibration optimization method based on hand–eye data filtering is proposed to improve the accuracy of hand–eye calibration;
2. An improved U-net segmentation method is proposed to accurately segment the depth image and achieve fast and accurate segmentation of point cloud;
3. The point cloud registration strategy of "SAC-IA coarse registration-ICP fine registration" is adopted to achieve the target pose acquisition.

The rest of the paper is arranged as the following: the Section 2 introduces the principle of hand–eye calibration and the method flow based on hand–eye calibration data filtering; the Section 3 describes point cloud segmentation and object 6D pose estimation based on deep learning; the Section 4 verifies the advantages and feasibility of the system; finally, the Section 5 summarizes the work of the paper and prospects future research issues.

2. Optimization of Hand–Eye Calibration Based on Data Filtering

2.1. Mathematical Model and Error Analysis of Hand–Eye Calibration

The hand–eye calibration problem is usually defined as the problem of solving the equation $AX = XB$. The Figure 2 shows the schematic diagram of hand–eye calibration. The hand–eye calibration matrix solution satisfies Formula (1).

$$\begin{array}{c} A_1XB_1 = A_2XB_2 \\ {A_2}^{-1}A_1X = XB_2{B_1}^{-1} \\ AX = XB \end{array} \tag{1}$$

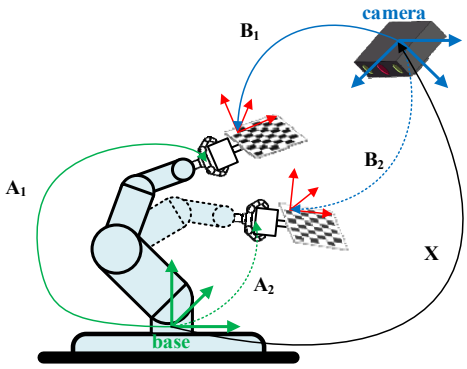


Figure 2. Schematic diagram of hand–eye calibration: (i = 1, 2) represents the transformation matrix from the robot base coordinate system to the robot end coordinate system; (i = 1, 2) represents the transformation matrix from the camera coordinate system to the calibration object coordinate system; X represents the transformation matrix from the robot base coordinate system to the camera coordinate system, that is, the matrix to be solved.

In the case of eye-to-hand, the positional relationship between the calibration object and the robot end remains unchanged. Therefore, according to multiple sets of hand–eye calibration data and the obtained X matrix, the transformation matrix P^b from the point p in the calibration object coordinate system to the robot terminal coordinate system is estimated multiple times. The accuracy of matrix X is judged by calculating the standard deviation of multiple sets of transformation matrices P^b. The smaller the standard deviation, the more accurate the matrix obtained. This error evaluation method is called the reprojection error analysis method. Formulas (2) and (3) respectively represent the solution formulas for matrix P^b and reprojection error E.

$$\begin{bmatrix} P^b \\ 1 \end{bmatrix} = {\overline{A}_i}^{-1} \cdot X \cdot \overline{B}_i \tag{2}$$

where P^b represents the transformation matrix from point p on the calibration object to the coordinate system of the robot end, $\overline{A}_i$ represents the transformation matrix of the i-th robot base coordinate system to the robot end coordinate system, $\overline{B}_i$ represents the transformation matrix from the i-th camera coordinate system to the calibration object coordinate system, X represents the hand–eye calibration matrix.

$$E = \sqrt{\frac{\sum_{i=1}^{n} (p_i - \overline{p})^2}{n}} \tag{3}$$

where, p_i represents the estimated value of the pose of the point p on the end of the robot on the i-th set of hand–eye calibration data, $\overline{p}$ represents the average of the pose estimation values of point p on the end of the robot on the n sets of calibration objects.

2.2. Hand–Eye Calibration Optimization Based on Data Filtering

Researchers have proposed various theories to solve the problem of hand–eye calibration $AX = XB$ [18–20]. Due to the calibration error of the camera's external parameters and the robot's own motion error, there will be errors in the hand–eye calibration solution, which is inevitable. The basic process of hand–eye calibration is shown in the Figure 3.

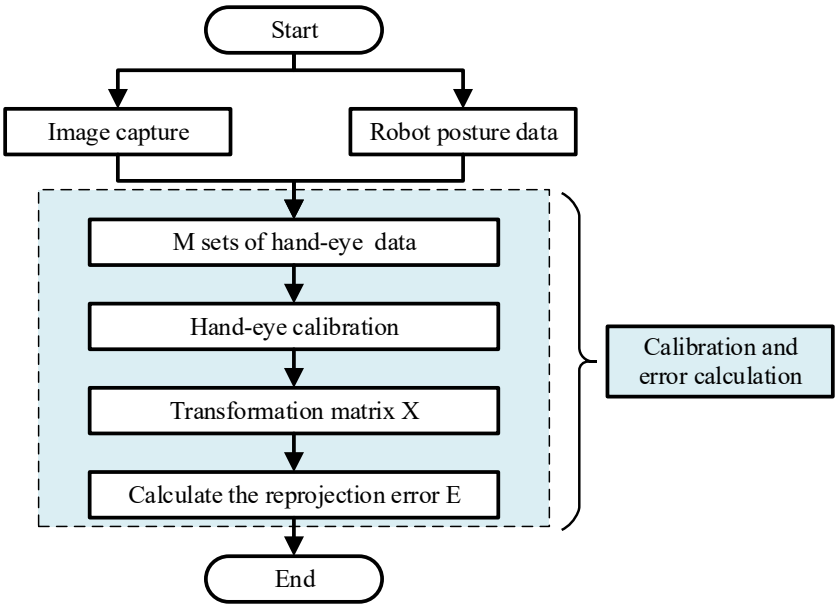


Figure 3. Hand–eye calibration process. During hand–eye calibration, M sets of A_i (i = 1, 2, . . . , M) and its response to B_i (i = 1, 2, . . . , M) is used to solve for X.

In the practical application of hand–eye calibration, improper selection of one or several sets of data will seriously affect the final calibration results. Aiming at the problem of excessive hand–eye calibration error caused by "bad" data, this paper proposes an optimization method. The method flow proposed in this paper is shown in Figure 4. Firstly, calculate the hand–eye calibration matrix X and its corresponding reprojection error E according to the M sets of initial data. Then, remove one set of data in turn and calculate the corresponding M sets of hand–eye calibration matrix X_i and the corresponding reprojection error E_i (i = 1, 2, . . . , M). Hand–eye calibration matrix X_i and corresponding reprojection error E_i are stored in array $X[\ ]$ and $E[\ ]$. The smallest element $E_{\min}$ in $E[\ ]$ is compared with E. If $E_{\min} \leq E$, then $E = E_{\min}$ and X is equal to the hand–eye calibration matrix $X[E_{\min}]$ corresponding to $E_{\min}$. This process only realizes the filtering of a single set of data. If you wish to realize the filtering of multiple sets of data, you can carry out this process several times. (Note: Data and matrix need to be updated.)

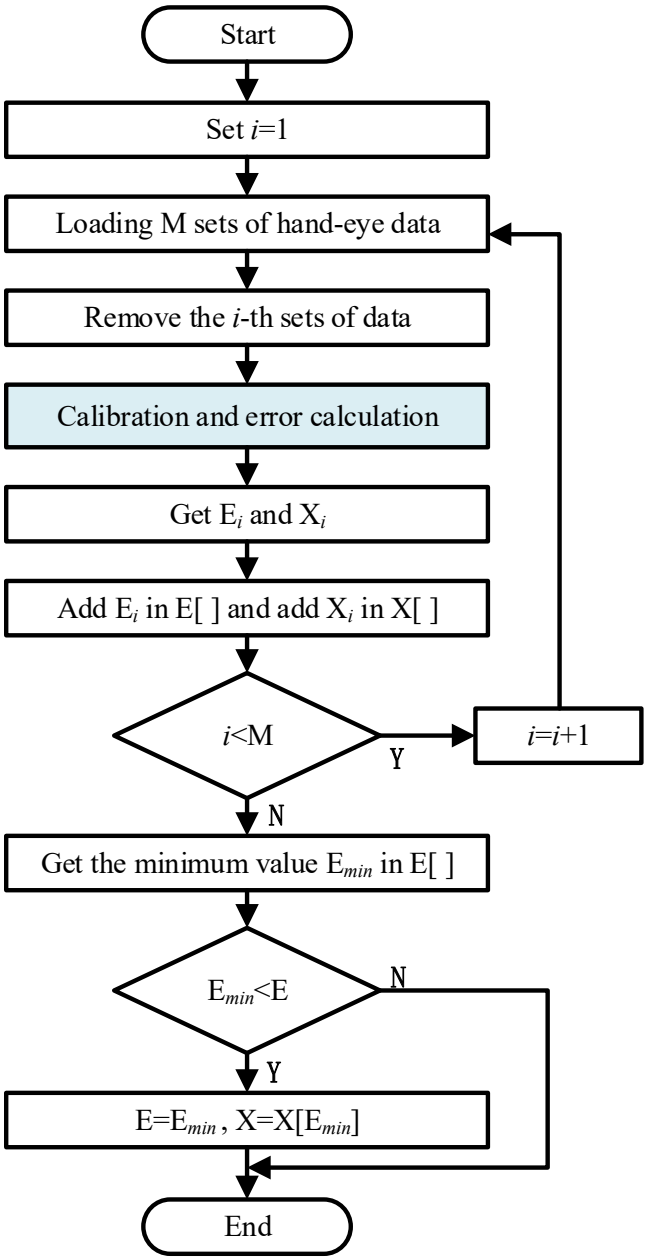


Figure 4. Hand–eye calibration optimization process based on data filtering.

3. Object Recognition and Pose Estimation

3.1. Object Segmentation and Recognition Based on Improved U-Net

Point cloud segmentation refers to the segmentation of data points with the same attributes and different attributes according to the relevant functional definition of the original point cloud data to obtain the classification of each point. In automatic assembly tasks based on 3D vision, accurate segmentation of point cloud is helpful to improve the efficiency and accuracy of high point cloud registration. Traditional methods use the shape, color, curvature and other features of a point cloud to classify them, but these methods are slightly inadequate in accuracy and robustness. Aiming at the difficulty of segmentation of 3D point cloud data, an improved U-net [21] is proposed to segment a 2D depth image to achieve segmentation of a target 3D point cloud. Compared with the existing 3D point cloud segmentation algorithm, this method performs segmentation on a 2D image without the complex preprocessing of point clouds and has a higher segmentation efficiency. As a classical deep learning model, U-net has the advantages of fewer training

samples, high segmentation accuracy, and being a lightweight model. In order to improve the segmentation accuracy of U-net, an attention module [22] is introduced in this paper to achieve accurate segmentation of object depth image.

The improved U-net structure in this paper is shown in Figure 5. The network is divided into two parts: a contracting path and an expansive path. The first layer uses a standard convolution module composed of two sets of 3 × 3 convolutions (string 1 and padding 1), batch normalization (BN), and ReLU activation to extract features. After down-sampling using max pooling, the second layer performs the aforementioned standard convolution operation. On the third and fifth layers of the network, in order to strengthen the feature extraction ability of the network, two sets of standard convolution operations are used after down-sampling. In the contracting path, the size of feature map is halved and the number of channels is doubled with each additional layer (the number of channels in the fifth layer remains unchanged). In the expansion path, in order to prevent over-fitting, the fifth layer is put into Dropout operation with a probability of 0.5. After up-sampling by bilinear interpolation, the feature maps from the fourth layer in the down-sampling stage are sent into the attention module together with the feature maps from the fourth layer in the down-sampling stage to output feature maps that enhance semantic information. The above operations are used on the sixth to the ninth layer of the network. With each additional floor, the size of the feature map is doubled and the number of channels is halved. For the feature map of the ninth layer, 1 × 1 convolution is used to reduce the channel number of the feature map to 1, and then sigmoid operation is performed to obtain the prediction probability map. In the prediction, the output probability map of layer nine is used to generate a binarized segmentation image with 0.5 as the threshold.

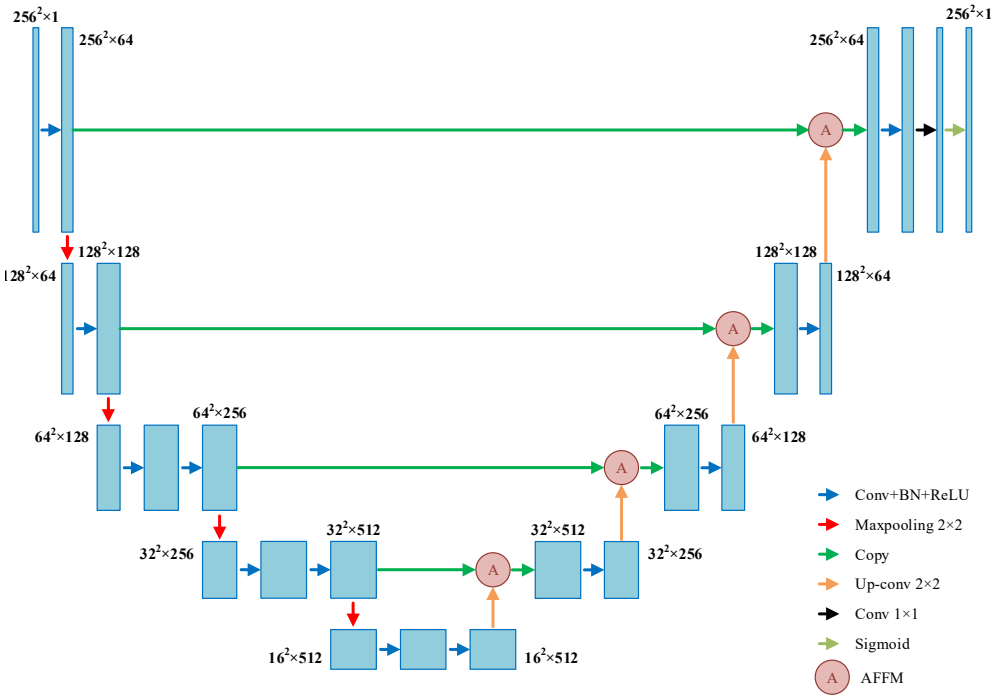


Figure 5. Improved U-net structure.

As shown in Figure 6, the attention feature fusion module (AFFM) constructed in this paper consists of two branches, which are the shallow feature map L from the contracting path and the deep feature map H from the expanding path. Shallow feature map L and deep feature map H focus and retain key features and spatial information through channel attention module (CAM) and spatial attention module (SAM). After completing residual calculation, the shallow feature map L is sent into the CAM. After deep feature map H is multiplied by shallow feature map L, residual calculation is completed and output feature map O is obtained.

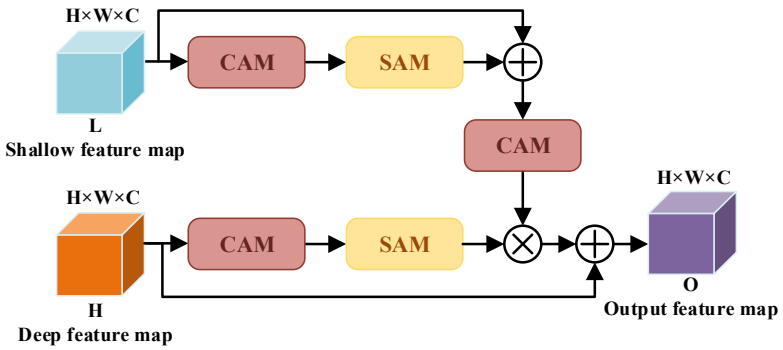


Figure 6. Attention feature fusion module.

AFFM mainly includes CAM and SAM, whose structures are shown in Figures 7 and 8 respectively.

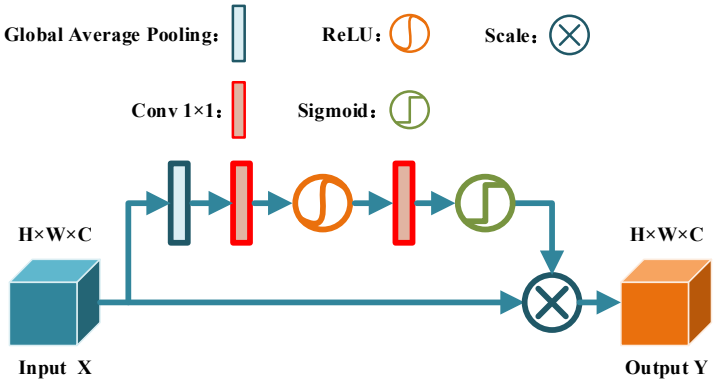


Figure 7. Channel attention module (CAM). In the CAM, firstly conduct Global Average Pooling (GAP) operation on input feature map X to obtain the Global information feature value of each channel feature map, and then conduct 1×1 convolution. Finally, the feature is transformed by ReLU activation, 1×1 convolution and Sigmoid operation to generate attention weight. The weight is multiplied by the input feature map X to obtain the output feature map Y, so as to achieve feature recalibration along the channel direction.

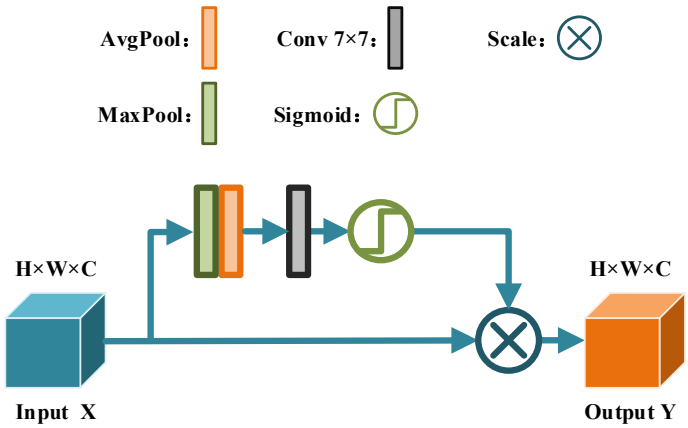


Figure 8. Spatial attention module (SAM). In the SAM, a maximum pooling and average pooling of the channel dimension are performed on the input feature map X to obtain two obtained channel descriptions. Splice these two descriptions together according to the channel. Then the features are nonlinearly transformed through 7×7 convolution and Sigmoid operation to generate attention weights. Finally, the attention weight is multiplied by the input feature map X to obtain the output feature map Y.

The image segmentation of assembly work is essentially a pixel-level binary classification task, and its loss function usually adopts the binary cross entropy loss (binary cross entropy loss, BCE Loss). In depth images, the number of target pixels is far less than the number of non-target pixels. If BCE is used as the loss function, the predicted results will be dominated by non-target pixels and the recognition ability will decrease. In addition, the target area in the image is often more difficult to identify than the non-target area. Therefore, in order to overcome these problems, this paper adopts the focus loss function, and its calculation formula is as follows:

$$\begin{aligned} F_L(p_t) &= -\alpha_t(1-p_t)^\gamma \log(p_t) \\ p_t &= \begin{cases} p & y=1 \\ 1-p & y=0 \end{cases} \end{aligned} \tag{4}$$

where p ($p \in [0,1]$) is the class probability output by the model, α_t is the loss weight of the t-th class sample, the sum of the loss weights of all classes is 1, and γ ($\gamma \geq 0$) controls the size of the loss of the difficult and easy samples. When γ increases, the model will pay more attention to samples that are difficult to distinguish; when $\gamma = 0$, the focus loss function degenerates into a normal cross entropy function with α_t.

3.2. Object Pose Estimation Based on "SAC-IA Coarse Registration–ICP Precise Registration"

Common point cloud registration algorithms include normal distribution transformation (NDT) [23], singular value decomposition (SVD) [24], iterative closest point (ICP) [25], and many improved algorithms. Among them, the principle of ICP algorithm is simple and easy to understand, and the registration effect is remarkable. But the ICP algorithm is very sensitive to the object's initial pose; a bad initial pose may lead ICP to converge in a wrong pose. To solve the problems of ICP, the point cloud registration strategy of "SAC-IA coarse registration–ICP precise registration" is adopted in this paper. As shown in Figure 9, it is the flow chart of the point cloud registration strategy used in this paper. Firstly, the Fast Point Feature Histogram Description (FPFH) [26] is used as the point cloud feature description, and the sampling consistent initial algorithm (SAC-IA) is used to coarsely register the point cloud. Through SAC-IA coarse registration, the ICP point cloud registration can obtain a good initial value and avoid falling into the local optimum. The k-d tree data structure is used to improve the query speed of the nearest neighbors of the ICP algorithm.

In the preprocessing phase, it is necessary to use a filtering algorithm to remove outliers and down-sample the point cloud. The statistical outlier removal method was used to remove the noise and outliers. The point cloud P_q^* after removing outliers and noise can be represented by Formula (5).

$$P_q^* = \left\{ P_q^* \in P \middle| (\mu_k - \sigma_k) \leq \overline{d} \leq (\mu_k + \sigma_k) \right\} \tag{5}$$

where $\overline{d}$ represents the mean distance from a point p_q in the point cloud P to the k nearest neighbors, μ_k and σ_k represents the mean and standard deviation of the Gaussian distribution of the average distance of the point cloud, respectively.

After the removal of outliers and noises, we used Voxel-Grid algorithm to down-sample the point cloud. The Voxel-Grid down-sampling algorithm builds multiple voxels based on the input size (each voxel is a set containing a different number of points). Then, the centroid of each voxel is calculated. Finally, the other points of the corresponding voxels are represented by the centroids. The calculation formula of the centroid is shown in Formula (6).

$$\mu_p = \frac{1}{m}\sum_{i=1}^{m} p_i \tag{6}$$

where p_i represents the points contained in a voxel, and m represents the number of points in the voxel.

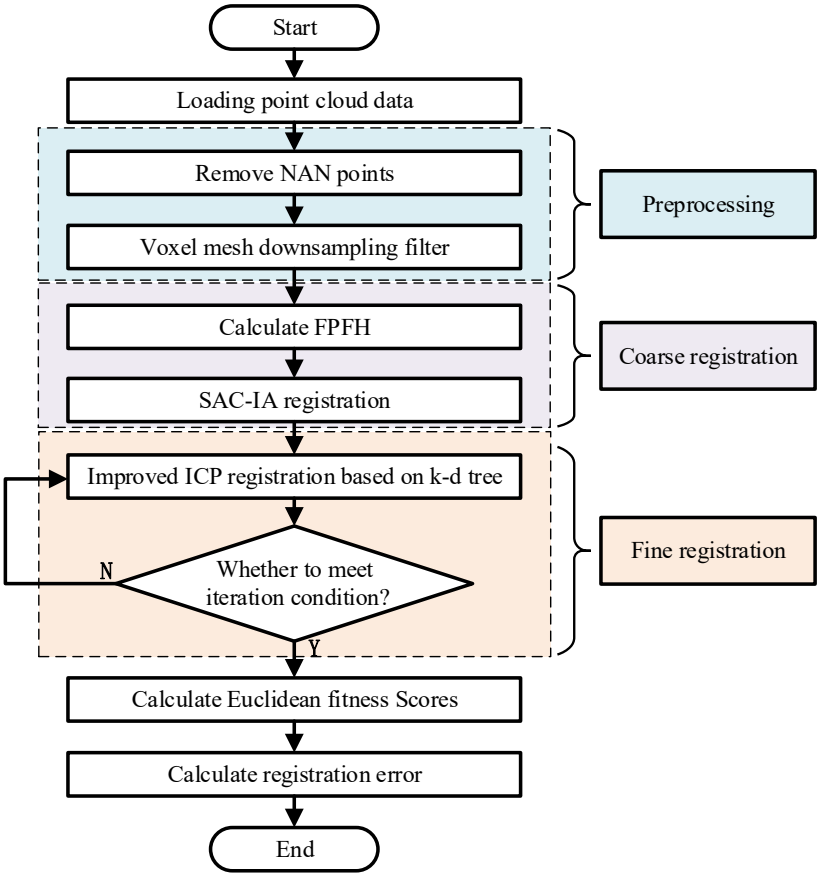


Figure 9. Point cloud registration flow chart.

The *FPFH* feature was first proposed in [26], which reduces the computational complexity of the algorithm to $O(nk)$ while still retaining most of the discriminative power of *PFH*. The *FPFH* feature diagram is shown in Figure 10. The expression of the *FPFH* feature is as follows:

$$FPFH(p) = SPF(p) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{w_k} \cdot SPF(p_k) \tag{7}$$

where the weight w_k represents the distance between query point p and a neighbor point p_k in a given metric space.

RANSAC is a common method to find the best match in cases where outliers are included. SAC-IA is a RANSAC-based algorithm for finding the best 3D rigid transformation matrix for 3D model registration. The algorithm flow is shown in Figure 11.

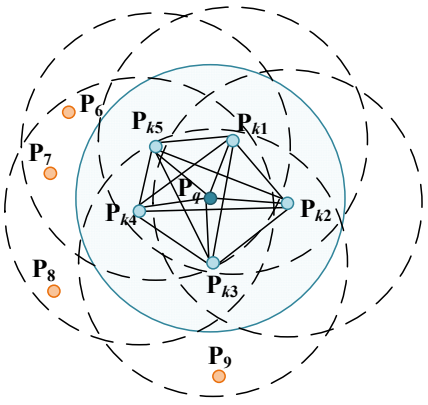


Figure 10. The *FPFH* feature diagram.

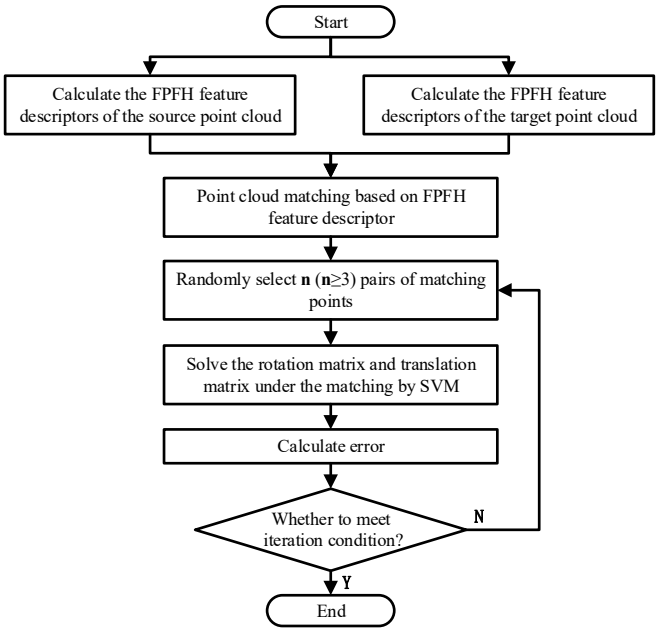


Figure 11. SAC-IA point cloud coarse registration based on *FPFH* features.

The ICP algorithm minimizes the distance between the point on the source point cloud and the corresponding point on the target point cloud through parameter update iteration. For the points in the source point cloud P, the corresponding closest point in the target point cloud Q is calculated by Euclidean distance. According to its corresponding relationship, solve the optimal transformation matrix (transition matrix R and displacement vector T). A new source point cloud P is obtained according to the transformation matrix, and the above process is iteratively executed until the convergence condition is satisfied. The objective function expression of the optimal transformation matrix is shown in Formula (8).

$$f(R,T) = \frac{1}{k}\sum_{i=1}^{k} \|q_i - (Rp_i + T)\|^2 \tag{8}$$

where the source cloud collection is $P = \{p_i | i = 1,2,3,...\}$, $Q = \{q_i | i = 1,2,3,...\}$.

4. Experiment and Discussion

The automatic robotic assembly system is shown in Figure 12. When the hand–eye calibration matrix is correct, the error of the rotation matrix has a negligible effect on the assembly, so the error is not discussed during the experiment.

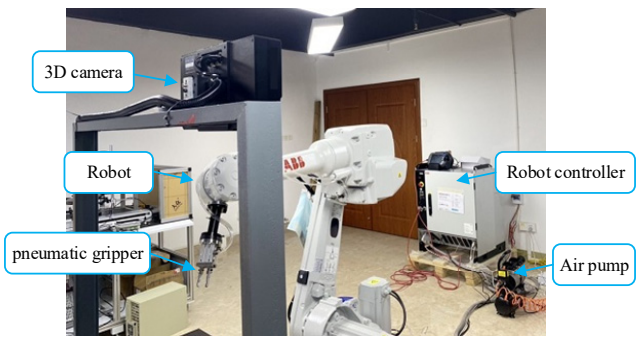


Figure 12. Robotic assembly system. In order to obtain high-quality point cloud data, we used a binocular structured light 3D camera with a point cloud resolution of up to 0.02 mm. Pneumatic grippers were installed at the end of the ABB IRB 2600-20 robot. The clamping state of the pneumatic gripper could be controlled by I/O programming. The robot was controlled by the robot controller.

4.1. Hand–Eye Calibration Experiment

In order to quantitatively evaluate the feasibility of the hand–eye calibration data filtering algorithm, we conducted 10 sets of repetitive experiments. The experimental platform is shown in Figure 13. In each experiment, 20 sets of hand–eye calibration data were collected and filtered out for 5 iterations. The error was calculated by Formula (3), and the result is shown in Figure 14. It can be seen from the experimental results that the hand–eye data filtering algorithm proposed in this paper reduces the reprojection error of the hand–eye calibration by 0.42 mm to 0.08 mm compared to the original data.

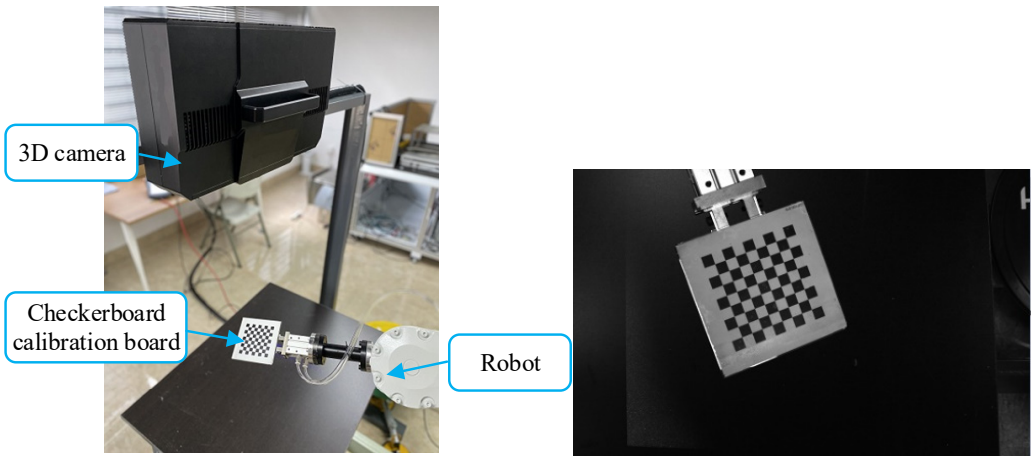


Figure 13. Hand–eye calibration data collection. The calibration board was installed to the end of the robot, and the image of the calibration board was collected through the camera.

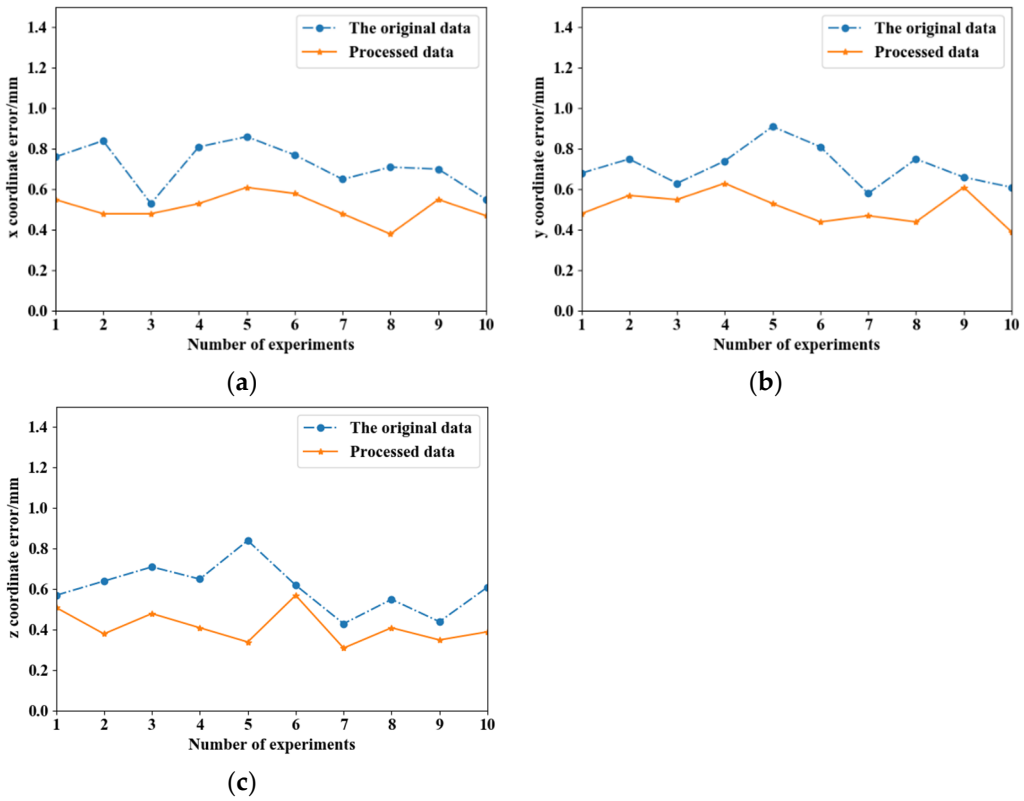


Figure 14. Reprojection error: (**a**) x coordinate reprojection error; (**b**) y coordinate reprojection error; (**c**) z coordinate reprojection error.

As shown in Figure 15, in order to further verify the feasibility of the proposed hand–eye calibration optimization algorithm, a point-to-point verification experiment was carried out. The error analysis between the transformation matrix and the actual thimble arrival pose was also carried out. The results of the hand–eye calibration error are shown in Figure 16. From the experimental results, the actual hand–eye calibration error was larger than the reprojection error. Through the hand–eye calibration optimization algorithm proposed in this paper, the hand–eye calibration error was reduced by 0.65 mm to 0.06 mm compared with the original data.

From the two verification experiments, the hand–eye calibration optimization method based on hand–eye data filtering proposed in this paper can improve the accuracy of hand–eye result. The method in this paper ensures the reliability of hand–eye calibration accuracy when non-professionals perform hand–eye calibration, which is very meaningful for industrial applications.

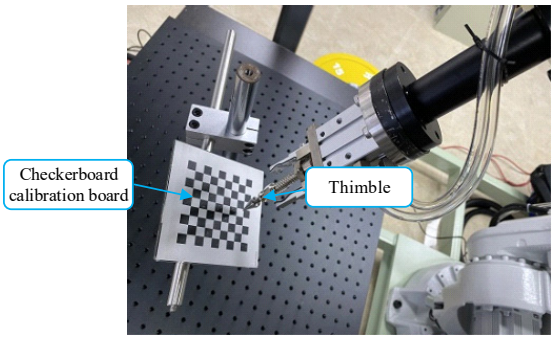


Figure 15. Point-to-point experiment. A thimble was installed at the end of the robot, and the transformation matrix from the point on the calibration board to the tip of the thimble was obtained according to the extracted pose of the calibration board relative to the camera coordinate system and the calculated hand–eye calibration matrix.

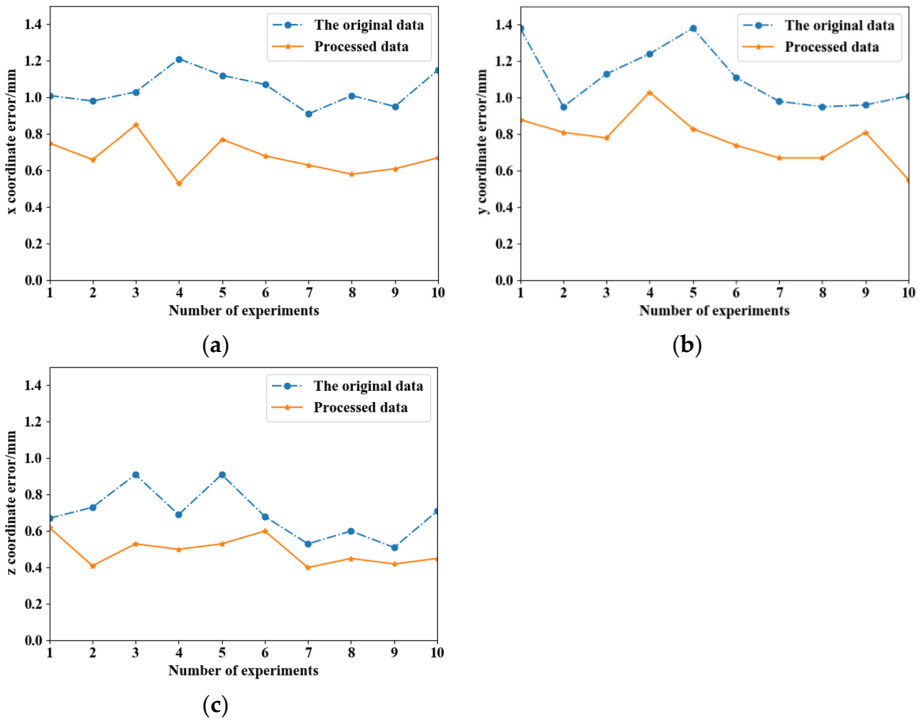


Figure 16. Hand–eye calibration error: (**a**) x coordinate hand–eye calibration error; (**b**) y coordinate hand–eye calibration error; (**c**) z coordinate hand–eye calibration error.

4.2. Pose Estimation and Assembly Experiment

In this paper, the Neutrik plug was used as the experimental object, and the binocular structured light 3D camera mentioned in this paper was used for data collection. During the data collection process, we set the aperture and exposure time of the camera and the projection light intensity of the projector to ensure that the collected data were under the same lighting environment (the experimental environment was indoors, so the influence of ambient light can be ignored). We collected 500 sets of Neutrik plug point cloud data with different poses and converted them into depth images. The image and depth image of the Neutrik plug are shown in Figure 17. After finishing the labeling of all image data, the data set was divided into a training set (400 sheets), a validation set (50 sheets), and a test set (50 sheets).

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

Figure 17. The image and depth image of the Neutrik plug: (**a**) the image of the Neutrik plug; (**b**) the depth image of the Neutrik plug.

All models were trained and tested in the environment of Windows10 + Python3.6 + Pytorch1.1, and are accelerated by an RTX 2060 graphics card with 16 GB memory. In this paper, an adaptive moment estimation optimizer (Adam) was used to update the network parameters iteratively. The batch size was 4 and the training was 50 epochs. The initial value of the learning rate was set to 0.001, and the ReduceLROnPlateau dynamic learning rate adjustment strategy was used to complete the automatic decay of the learning rate.

In order to quantitatively evaluate the performance of the improved U-net image segmentation in this paper, accuracy (A_{cc}) and Dice coefficient (D_{ice}) were used as evaluation indicators. The calculation formula is as follows:

$$A_{cc} = \frac{T_P + T_N}{T_P + T_N + F_N + F_P} \tag{9}$$

$$D_{ice} = \frac{2T_P}{2T_P + F_P + F_N} \tag{10}$$

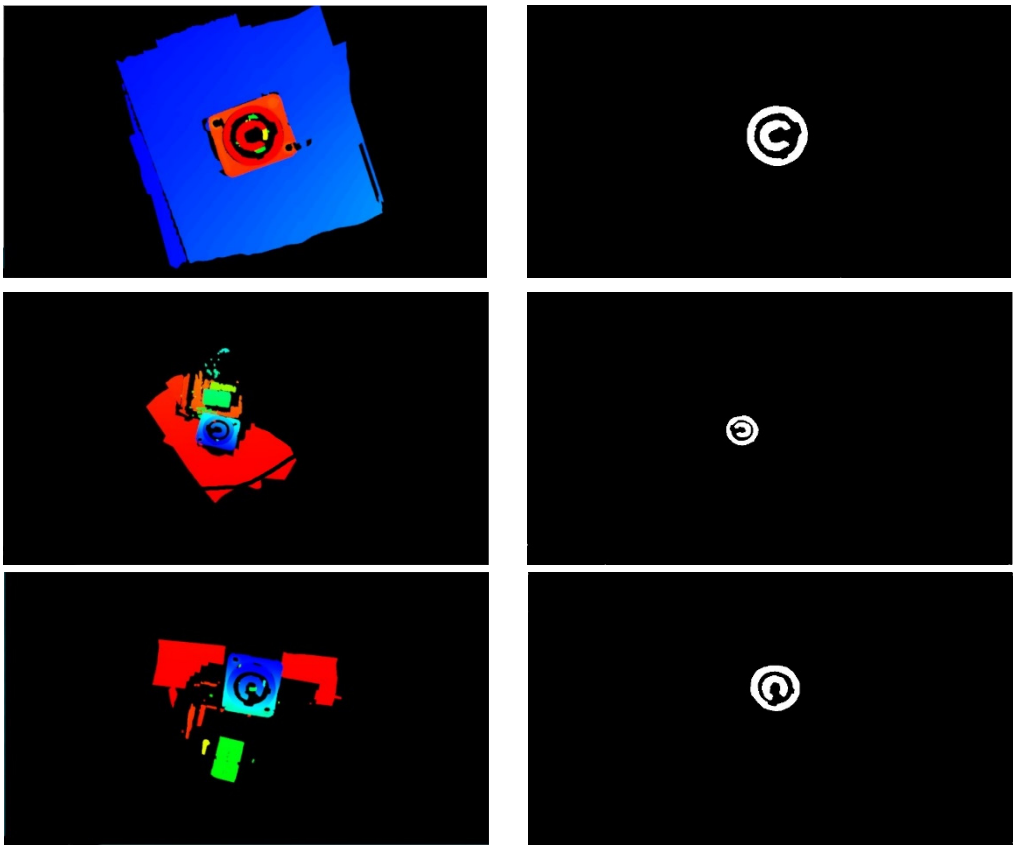
where T_P is the number of pixels that are actually the target area and accurately recognized as the target area. F_N is the number of pixels that are actually the target area but are recognized as non-target areas. T_N is the actual non-target area and accurately recognized as non-target areas. F_P is the number of pixels that are actually not the target area but are recognized as the target areas. The value range of the above indicators is between 0 and 1. The larger the value, the better the model performance.

As shown in Table 1,comparing the improved U-net in this paper with U-net, Attention U-net [27], R2U-net [28] and DeepLab V3+ [29], the results show that the A_{cc} coefficient and D_{ice} coefficient of the method in this paper reached 0.961 and 0.876, respectively, which is better than the other four methods.

Table 1. Performance comparison of image segmentation of different deep learning models.

Methods	A_{cc}	D_{ice}
U-net	0.921	0.813
Attention U-Net [27]	0.933	0.825
R2U-Net [28]	0.937	0.831
DeepLab V3+ [29]	0.944	0.843
Our Method	0.961	0.876

As shown in Figure 18, it is the Neutrik plug depth image and the segmented image using the method in this paper. From the segmentation results, the improved U-net proposed in this paper can achieve point cloud segmentation for specific regions.



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

Figure 18. Neutrik plug depth image andsegmented image: (**a**) Neutrik plug depth image; (**b**) segmented image.

We used the point cloud registration method proposed in this paper to perform point cloud registration on the segmented point cloud data, and compare the registration results with other methods. The comparison results are shown in Table 2. It can be seen from the table that although the method proposed in this paper takes more time overall than Seget-ICP [30] and BiLuNetICP [31], the registration accuracy is improved. For assembly tasks, the increase in registration accuracy is conducive to the stability of the assembly process, so the method in this paper is valuable and meaningful.

Table 2. 6D Pose Estimation Results.

Methods	Rotation Mean Error (°)	Translation Mean Error (mm)	Runtime (s)
Seget-ICP [30]	1.55	1.27	6.5
BiLuNetICP [31]	1.32	1.25	6.8
PoseCNN + ICP [32]	1.29	1.21	11.5
GO-ICP [33]	1.35	1.33	16.7
Our Method	1.27	1.19	7.5

The effect of point cloud registration using the method proposed in this paper is shown in Figure 19. As shown in the Figure 20, in order to verify the feasibility of the automatic assembly system proposed in this paper, a verification platform is built. The female Neutrik plug (Workpiece B) was fixed under the 3D camera, and the male Neutrik plug (Workpiece A) was clamped by the robot in a designated posture.

Figure 19. The effect of point cloud registration using the method proposed in this paper.

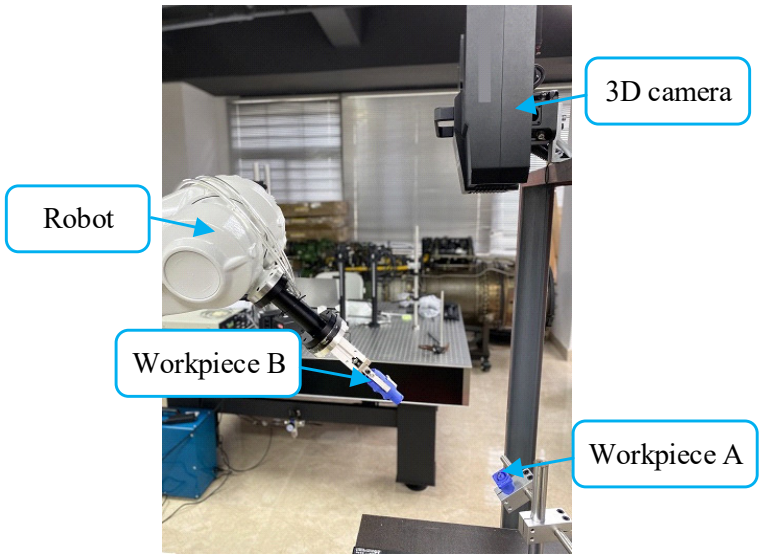


Figure 20. Automatic assembly system experiment.

We compared the posture calculated by the automatic assembly system with the real posture and analyzed the error. The error of the experiments is shown in Figure 21. The experimental results show that the assembly error of the automatic assembly system in this paper is between 0.7~1.5 mm, which meets the requirements of Neutrik plug assembly.

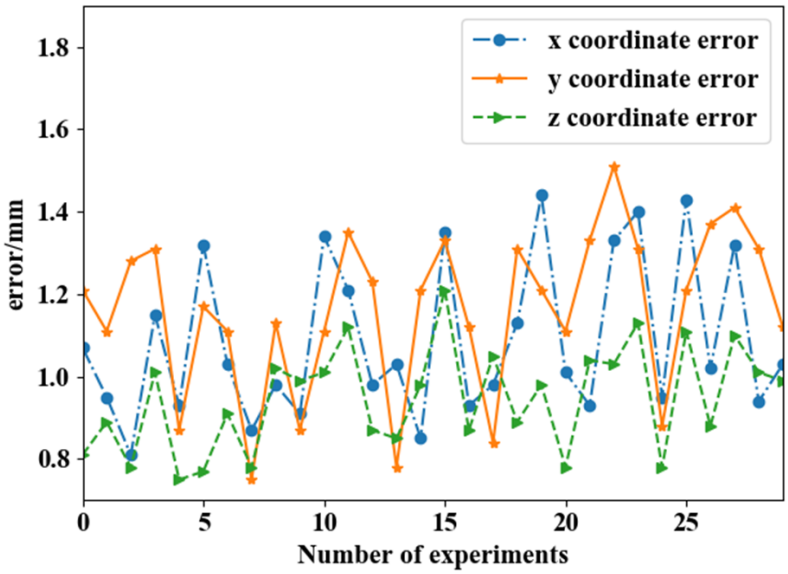


Figure 21. Assembly error analysis diagram.

5. Conclusions

The experiment results show that the general automatic assembly system based on 3D vision proposed in this paper is feasible. Through the hand–eye calibration method based on data filtering, the coordinates of the vision sensor and the coordinates of the robot are more accurately correlated. The improved U-net is used for image segmentation, which solves the issue of separating the target point cloud from the background. Compared with the traditional point cloud segmentation method, this method is more efficient. "SAC-IA coarse registration–ICP fine registration" is adopted to ensure the accuracy and efficiency of point cloud registration.

Of course, for the research of automatic assembly technology, the research in this paper still has some shortcomings. In the hand–eye calibration work, we directly obtained the motion parameters of the robot without compensating for the error disturbance of the robot motion parameters. Using inertial measurement units (IMU) to obtain the position of the robot is a common and effective means [34,35]. In future work, we will try to use IMU to obtain the motion parameters of the robot, which may help to further improve the accuracy of hand–eye calibration. In this paper, we used a deep-learning algorithm to improve the efficiency and accuracy of point cloud registration, but it needs to collect a large number of data samples and perform sample processing in advance. We need to further study the method of training deep learning based on small data samples to reduce the time cost.

Author Contributions: Conceptualization, P.L. and G.L.; methodology, P.L.; software, W.L.; validation, C.Z.; formal analysis, P.L., G.L. and C.Z.; investigation, P.L.; resources, W.L. and C.Z.; data curation, C.Z.; writing—original draft preparation, C.Z., P.L. and G.L.; writing—review and editing., C.Z. and W.L.; visualization, P.L.; supervision, W.L. and C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China under Grant no.2018YFB1305703 and Scientific and Technological Program of Quanzhou City under Grant no. 2019CT009.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

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

References

1. Peng, G.L.; Ji, M.Y.; Xue, Y.; Sun, Y. Development of a novel integrated automated assembly system for large volume components in outdoor environment. *Measurement* **2021**, *168*, 108294. [CrossRef]
2. Wang, S.; Chen, G.D.; Xu, H.; Wang, Z. A robotic peg-in-hole assembly strategy based on variable compliance center. *IEEE Access* **2019**, *7*, 167534–167546. [CrossRef]
3. Zeng, F.; Xiao, J.L.; Liu, H.T. Force/torque sensorless compliant control strategy for assembly tasks using a 6-DOF collaborative robot. *IEEE Access* **2019**, *7*, 108795–108805. [CrossRef]
4. Gai, Y.H.; Guo, J.M.; Wu, D.; Chen, K. Feature-Based Compliance Control for Precise Peg-in-Hole Assembly. *arXiv* **2020**, arXiv:2021.3112990. [CrossRef]
5. Park, H.; Park, J.; Lee, D.-H.; Park, J.-H.; Bae, J.-H. Compliant peg-in-hole assembly using partial spiral force trajectory with tilted peg posture. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4447–4454. [CrossRef]
6. Ma, Y.Q.; Du, K.; Zhou, D.F.; Zhang, J.; Liu, X.L.; Xu, D. Automatic precision robot assembly system with microscopic vision and force sensor. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 172988141985161. [CrossRef]
7. Li, C.; Chen, P.; Xu, X.; Wang, X.Y.; Yin, A.J. A Coarse-to-Fine Method for Estimating the Axis Pose Based on 3D Point Clouds in Robotic Cylindrical Shaft-in-Hole Assembly. *Sensors* **2021**, *21*, 4064. [CrossRef]
8. Liu, S.; Xing, D.-P.; Li, Y.-F.; Zhang, J.W.; Xu, D. Robust insertion control for precision assembly with passive compliance combining vision and force information. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 1974–1985. [CrossRef]
9. Qin, F.B.; Xu, D.; Zhang, D.P.; Li, Y. Robotic skill learning for precision assembly with microscopic vision and force feedback. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 1117–1128. [CrossRef]
10. Song, R.; Li, F.M.; Quan, W.; Yang, X.T.; Zhao, J. Skill learning for robotic assembly based on visual perspectives and force sensing. *Robot. Auton. Syst.* **2021**, *135*, 103651. [CrossRef]
11. Wang, H.S.; Ni, H.; Wang, J.C.; Chen, W.D. Hybrid Vision/Force Control of Soft Robot Based on a Deformation Model. *IEEE Trans. Control Syst. Technol.* **2019**, *29*, 661–671. [CrossRef]

12. Chang, W.C.; Weng, Y.H.; Tsai, Y.H.; Chang, C.L. Automatic robot assembly with eye-in-hand stereo vision. In Proceedings of the 2011 9th World Congress on Intelligent Control and Automation, Taipei, China, 21–25 June 2011; pp. 914–919.
13. Jiang, T.; Cui, H.H.; Cheng, X.S. A calibration strategy for vision-guided robot assembly system of large cabin. *Measurement* **2020**, *163*, 107991. [CrossRef]
14. Dong, D.W.; Yang, X.S.; Hu, H.P.; Lou, Y.J. Pose estimation of components in 3C products based on point cloud registration. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 20 January 2020; pp. 340–345.
15. Yan, S.H.; Tao, X.; Xu, D. High-precision robotic assembly system using three-dimensional vision. *Int. J. Adv. Robot. Syst.* **2021**, *18*, 172988142110270. [CrossRef]
16. Litvak, Y.; Biess, A.; Bar-Hillel, A. Learning Pose Estimation for High-Precision Robotic Assembly Using Simulated Depth Images. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 3521–3527.
17. Li, C.H.G.; Chang, Y.M. Automated visual positioning and precision placement of a workpiece using deep learning. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 4527–4538. [CrossRef]
18. Bedaka, A.K.; Lee, S.C.; Mahmoud, A.M.; Cheng, Y.S.; Lin, C.-Y. A Camera-Based Position Correction System for Autonomous Production Line Inspection. *Sensors* **2021**, *21*, 4071. [CrossRef]
19. Qiu, S.W.; Wang, M.M.; Kermani, M.R. A New Formulation for Hand–Eye Calibrations as Point-Set Matching. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 6490–6498. [CrossRef]
20. Hua, J.; Zeng, L.C. Hand–Eye Calibration Algorithm Based on an Optimized Neural Network. *Actuators* **2021**, *10*, 85. [CrossRef]
21. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
22. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
23. Biber, P. The normal distributions transform: A new approach to laser scan matching. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), Las Vegas, NV, USA, 27–31 October 2003; pp. 2743–2748.
24. Arun, K.S.; Huang, T.S.; Blostein, S.D. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *9*, 698–700. [CrossRef]
25. Besl, P.J.; Mckay, H.D. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [CrossRef]
26. Rusu, R.B.; Blodow, N.; Beetz, M. Fast point feature histograms (FPFH) for 3D registration. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 3212–3217.
27. Oktay, O.; Schlemper, J.; Folgoc, L.L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; Mcdonagh, S.; Hammerla, N.Y.; Kainz, B. Attention u-net: Learning where to look for the pancreas. *arXiv* **2018**, arXiv:1804.03999.
28. Alom, M.Z.; Yakopcic, C.; Taha, T.M.; Asari, V.K. Nuclei segmentation with recurrent residual convolutional neural networks-based U-Net (R2U-Net). In Proceedings of the NAECON 2018-IEEE National Aerospace and Electronics Conference, Dayton, OH, USA, 23–26 July 2018; pp. 228–233.
29. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
30. Wong, J.M.; Kee, V.; Le, T.; Wagner, S.; Mariottini, G.L.; Schneider, A.; Hamilton, L.; Hebert, M.; Johnson, D.M.S.; Wu, J.; et al. Segicp: Integrated deep semantic segmentation and pose estimation. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 5784–5789.
31. Van-Tran, L.; Lin, H.Y. BiLuNetICP: A Deep Neural Network for Object Semantic Segmentation and 6D Pose Recognition. *IEEE Sens. J.* **2020**, *21*, 11748–11757. [CrossRef]
32. Xiang, Y.; Schmidt, T.; Narayanan, V.; Fox, D. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv* **2017**, arXiv:1711.00199.
33. Yang, J.; Li, H.; Campbell, D.; Jia, Y.D. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 241–2254. [CrossRef]
34. Glowinski, S.; Obst, M.; Majdanik, S.; Potocka-Banas, B. Dynamic Model of a Humanoid Exoskeleton of a Lower Limb with Hydraulic Actuators. *Sensors* **2021**, *21*, 3432. [CrossRef]
35. Campos, B.A.N.; Motta, J.M.S.T. Online Measuring of Robot Positions Using Inertial Measurement Units, Sensor Fusion and Artificial Intelligence. *IEEE Access* **2021**, *9*, 5678–5689. [CrossRef]