The Pose Adjustment System of Robotic Arm Adopts Binocular Vision and Machine Learning

Yabing Ren[1], Hongyang Yu[2]

[1,2]Research Institute Electronic Science and TechnologyUniversity of Electronic Science and Technology of China, Chengdu, China

hyyu@uestc.edu.cn

Abstract. In order to solve the problem of rapid positioning of doors, windows and frames existing in the plastering process of the intelligent plastering robot, a pose adjustment system of robotic arm based on binocular vision and machine learning was proposed. The system tested a variety of feature point detection algorithms, description algorithms and matching algorithms, and proposed a combination scheme with good detection speed, feature point number and match accuracy: SURF algorithm and FLANN matching algorithm, and added mismatching filtering and density control strategy. The innovative method of using machine learning neural network to learn the reconstructed 3D point cloud improves the system's fault tolerance ability and response speed, and can quickly convert the complexly point cloud information into the attitude adjustment information required by the robotic arm. The experimental results show that the system to a single location within 2 s time, including robotic arm action time of 0.3 s to 0.5 s, after 1 to 4 times of circulation adjustment, can achieve the displacement error of 1 mm and 1 ° rotation error. The system can meet the requirements of accuracy, time and stability in the actual working environment.

Keywords: Binocular vision, visual measurement, machine learning, neural network, robotic arm, intelligent plastering robot.

1. Introduction

With the continuous improvement of binocular vision theory and algorithms, visual systems are increasingly applied in various fields, such as various home robots, service robots, measurement systems, and face recognition systems. However, in the field of architecture, the research and application of intelligent building robots are relatively few, especially in the aspect of building plaster, there is no intelligent equipment that can replace artificial. In the traditional construction, on the one hand, the mortar and construction process used in different positions of the wall are different, which makes the plastering work of the wall complex and heavy. On the other hand, due to the uneven skill level of plastering workers, it is difficult to control the engineering quality of wall plastering, which will have a greater impact on the subsequent process. Therefore, the traditional building field is in urgent need of innovation and upgrading and intelligent building robot is a very promising direction. The theoretical framework of computer vision was proposed by Marr[1] in the late 1970s. It used two two-dimensional images with parallax to generate three-dimensional images with depth information, which laid the theoretical foundation for the development of stereo vision. In the past few decades, the computer vision theory has been developing continuously. In 2006, Herbert Bay proposed SURF[2] algorithm, which is a more efficient and robust local feature point detection and description algorithm. After entering the 21st century, with the increasingly in-depth research on machine learning, it has become a core research direction in the field of artificial intelligence.

The pose adjustment system of robotic arm described in this paper is used to solve the positioning problem of the wall (door, window, frame, etc.) of the intelligent erasing robot platform. The system consists of binocular vision, machine learning neural network and robotic arm server. The binocular vision subsystem can quickly extract and match feature points and calculate 3D point cloud information from the left and right images, and then by the trained neural network model of point cloud information judgment and posture information is given, and the last action performed by the

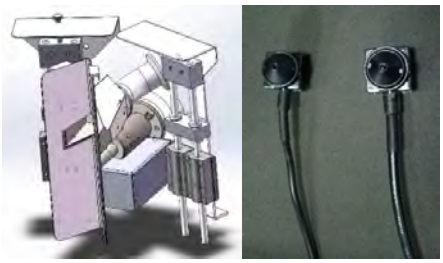
robotic arm target. The system can accomplish the pose adjustment of the intelligent plastering robot in a quick, accurate and stable manner.

2. The Experimental System

The experimental platform is an intelligent plastering robot independently developed by the research team. The experimental system involved in this paper is mainly composed of the following parts: the robotic arm, the end-effector, two parallel ordinary cameras, LED lighting and the master computer, etc. Fig. 1 is a schematic diagram of the various components of the system.

The robotic arm adopts the Swedish ABB IRB1200 model, which is installed horizontally on the linear lifting mechanism of the intelligent plastering robot. The end-effector includes a connecting base, a binocular camera base, a stroke trigger mechanism and various types of eraser. The camera is a common cone camera with a focal length of 3.7mm and a viewing angle of 60°. The master computer is configured with window10 system, i7 processor, 16G memory and 500G storage capacity.

(a) experiment platform (b) robotic arm



(c) end-effector (d) binocular camera

Fig. 1 Composition of the experimental system

3. Experimental Principle and Method

3.1 Parallel Binocular Vision Model

As shown in Fig. 2, the parallel binocular vision model is measured according to the parallax principle of images[3]. There is a pixel difference between the coordinates of the corresponding points of the same target point on the left and right camera imaging surfaces, so the 3D coordinates of the target point can be calculated by similarity triangle relationship.

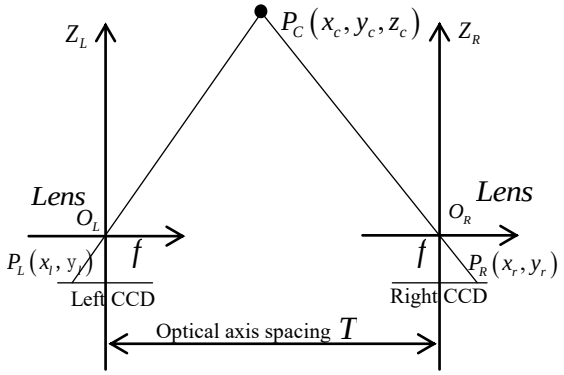


Fig. 2 Geometric model of parallel binocular vision

The camera coordinate system of the parallel binocular vision model is that X-axis from O_L to O_R, Z-axis from Z_L, and Y-axis direction according to the right hand rule. The world coordinate system is the camera coordinate system translation $T/2$ along the X direction, that is, the world coordinate system and the camera coordinate system have the following transformation relationship:

$$\begin{cases} x_w = x_c - \dfrac{T}{2} \\ y_w = y_c \\ z_w = z_c \end{cases} \tag{1}$$

In Fig. 2, $P_L(x_l, y_l)$ and $P_R(x_r, y_r)$ are respectively the coordinates of point P_C in the camera coordinate system on the image coordinate system of the left and right cameras. Therefore, from the similarity triangle relationship, we can get:

$$\begin{cases} x_l = f\dfrac{x_c}{z_c} \\ x_r = f\dfrac{x_c - T}{z_c} \\ y_l = y_r = f\dfrac{y_c}{z_c} \end{cases} \tag{2}$$

In addition, the relation between image coordinate system and pixel coordinate system is as follows:

$$\begin{cases} u = \dfrac{x}{dx} + u_0 \\ v = \dfrac{y}{dy} + v_0 \end{cases} \tag{3}$$

Where (u_0, v_0) is the principal point coordinating of the image. Since the left and right cameras in the system adopt cameras with the same parameter specifications, it is considered that dx_l, dy_l and dx_r, dy_r are the same. Finally, combining equations (1), (2) and (3), the conversion formula from the pixel coordinate system to the world coordinate system can be obtained:

$$\begin{cases} x_w = \dfrac{T}{2}\dfrac{(u_l - u_{l0}) + (u_r - u_{r0})}{(u_l - u_{l0}) - (u_r - u_{r0})} \\ y_w = T\dfrac{f_x}{f_y}\dfrac{((v_l - v_{l0}) - (v_r - v_{r0}))}{((u_l - u_{l0}) - (u_r - u_{r0}))} \\ z_w = \dfrac{T f_x}{(u_l - u_{l0}) - (u_r - u_{r0})} \end{cases} \tag{4}$$

According to equation (4), Just take the internal parameters $u_0, v_0, f_x = f/d_x, f_y = f/d_y$ of the left and right cameras, and then the three-dimensional coordinates of P_w can be obtained from the pixel coordinates (u_l, v_l) and (u_r, v_r) to achieve visual measurement.

3.2 Camera Calibration

The purpose of calibrating the binocular camera is to obtain the internal parameters u_0, v_0, f_x, f_y and external parameters of the left and right cameras, so as to realize the binocular vision measurement function. First, the left and right cameras were calibrated separately to obtain the internal and external parameters of the left and right cameras. Then, the binocular vision system is calibrated as a whole to obtain the rotation matrix R_L, R_R and translation matrix T_L, T_R of the left and right cameras. Finally, the rotation matrix and translation matrix of the right camera relative to the left camera are calculated, that is, the rotation matrix R_W and translation matrix T_W of the entire binocular vision system.

The coordinates of point P in the left and right camera coordinate system P_L, P_R and the world coordinate system P_W are known as follows:

$$\begin{cases} P_L = R_L P_W + T_L \\ P_R = R_R P_W + T_R \end{cases} \tag{5}$$

Therefore, the rotation matrix R_W and translation matrix T_W of the whole binocular vision system are as follows:

$$\begin{cases} R_W = R_L R_R^{-1} \\ T_W = T_L - R_L R_R^{-1} T_R \end{cases} \tag{6}$$

3.3 Image Feature Point Detection and Matching

The SIFT[4-5] algorithm is proposed by David G.Lowe based on the scale space. Its significant advantage is that it can maintain a certain invariance to the changes in the scale and rotation of the image, but the process of feature point extraction is very large due to its huge feature calculation. SURF algorithm improved the SIFT algorithm by using a more efficient feature extraction[6] and description method. It used Hessian matrix to extract feature points, harr wavelet statistics to allocate principal directions, and cut the SURF feature descriptor into half of SIFT.

The Hessian[7] matrix is actually the second-order partial derivative of the multivariate function, which uses the local curvature of the function to discriminate the corner points. For an image $f(x,y)$, its Hessian matrix is as follows:

$$H\left(f(x,y)\right) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial x \partial y} & \dfrac{\partial^2 f}{\partial y^2} \end{bmatrix} \tag{7}$$

In the image represented by a numerical matrix, the derivative is equivalent to the difference value of the adjacent pixel points, and then the Hessian matrix can be expressed by the following formula.

$$\det(H) = D_{xx} D_{yy} - D_{xy}^2 \tag{8}$$

In addition, before constructing the Hessian matrix, the SIFT algorithm processed images with Gaussian filtering, while SURF algorithm used a Box filter, as shown in Fig. 3. The Box filter converted the convolution calculation to the search and calculation of integral graph[8], greatly simplifying the filtering operation. In order to compensate for the errors caused by the replacement filter, SURF algorithm introduced a weighted coefficient w (empirical value of 0.9), then the Hessian matrix can be expressed by the following formula.

$$\det(H) = D_{xx} D_{yy} - \left(w D_{xy}\right)^2 \tag{9}$$

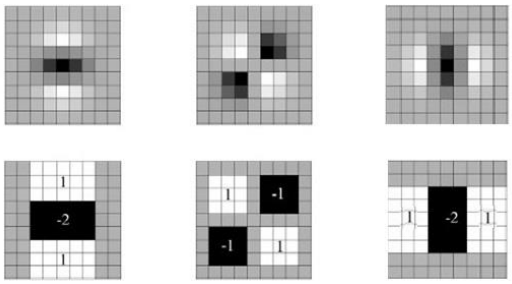


Fig. 3 Gaussian filter and Box filter

In the direction distribution of feature points, SUFR used harr wavelet statistics instead of SIFT gradient histogram statistics. The specific approach is to divide the circular neighborhood around the feature points into 6 sector regions, and then calculate the harr wavelet sign in each sector region, and finally assign the main direction according to the statistical results. Both SURF and SIFT express the matching degree of feature points by the length of the Euclidean distance between feature points.

3.4 Neural Network in Machine Learning

Neural network[9-11] in machine learning is a large and interdisciplinary field. "A neural network is a network of widely parallel interconnected by adaptive simple units whose organization can simulate the interaction of biological nervous systems with real-world objects" [Kohonen, 1988]. The BP neural network is used in the system, which is essentially a multi-layer feedforward neural network trained by the error back propagation algorithm.

As shown in Figure 4, it is a typical double hidden layer feedforward neural network, where L_1 is the input layer, L_2, L_3 is the hidden layer, and L_4 is the output layer. In the feedforward neural netwo rk, each node in layer $L_i, (i>1)$ will have a weight vector $W_{l_i,j}\left(w_1, w_2 \cdots, w_{N(l_{i-1})}\right),\ 0 < j <= N\left(L_i\right)$ correspondin g to the weight of $N\left(L_i\right)$ nodes in layer L_{i-1}, then $L_i = f\left(W_{l_i} \square L_{i-1}\right) + B_{l_i}$, where B is an offset vecto r and $f\left(x\right)$ is an activation function.

Activation function is an important concept in neural network, whose function is to simulate bi ological neurons to process the underlying signals, common activation functions are $sgn\left(x\right)$、$sigmoid\left(x\right)$、$relu\left(x\right)$、$\tanh\left(x\right)$ et al.

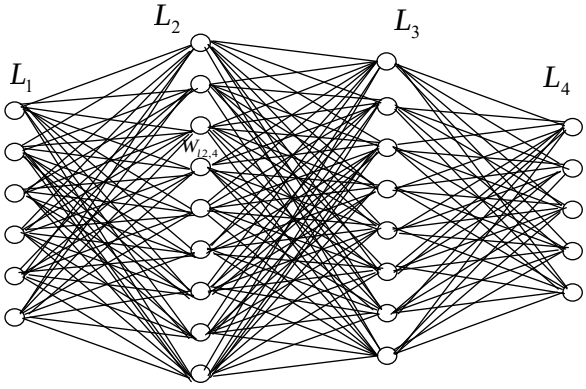


Fig. 4 Double hidden layer feedforward neural network

The self-learning of the feedforward neural network is realized by the Back Propagation algorithm, referred to as the BP algorithm. Most BP algorithms use a gradient descent training strategy that dynamically adjusts the parameters of the neural network with the target's negative gradient direction and a given learning rate η. The ultimate goal is to minimize the cumulative error on the training set.

4. Experiment and Analysis

4.1 Overall Process of Experiment

The pose adjustment system of robotic arm is mainly divided into two parts: the learning system and the working system.

The two systems share a binocular vision subsystem, as shown in Fig. 5. The overall process of the system experiment mainly includes camera calibration, binocular vision subsystem construction, neural network training and preparation of a variety of robot arm server.

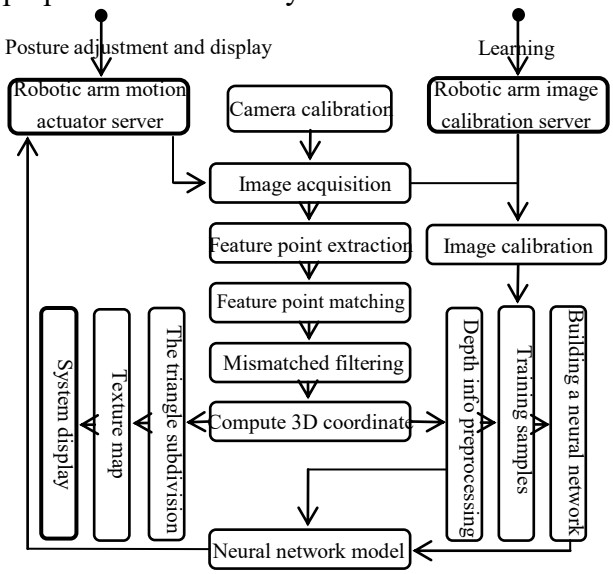


Fig. 5 Composition of the pose adjustment system of robotic arm

4.2 Camera Calibration and Mismatching Filtering

As for camera calibration, it is mainly to determine the internal parameters of left and right cameras. This experiment uses the Zhang Zhengyou[12] checkerboard calibration method with the advantages of flexibility and accuracy, and realizes the camera calibration program based on OpenCV. As shown in figure 6, is the detection result of inner corner point of the calibration image.

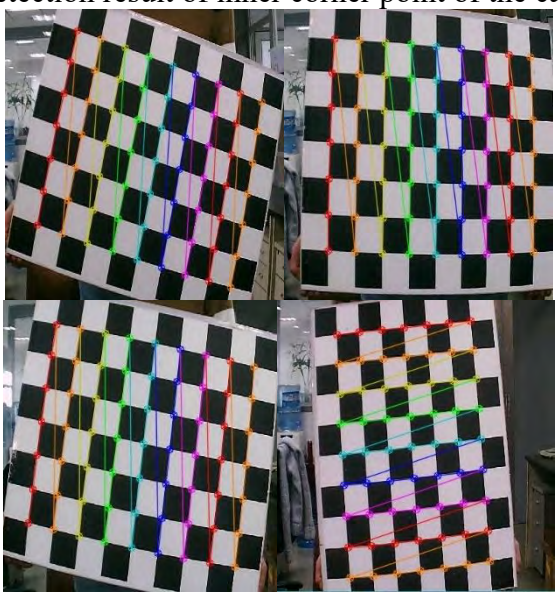


Fig. 6 Inner corner detection results of camera calibration image

As shown in Table 1, is the calibration result of left and right cameras:

Table 1 Internal parameters of the camera

Camera internal parameters(pixel)	Calibration error	u_0	v_0	f_x	f_y
Left camera	0.0453821	313.5579418	251.3065549	720.8317678	994.2579618
Right camera	0.0299798	320.4305297	240.4909714	704.2199450	940.7176773

As for the mismatched filtering algorithm, this paper has made relevant introduction in the principle section. On the one hand, due to the inevitable relative rotation and translation of the imaging plane of the left and right cameras, on the other hand, the feature point matching algorithm also has mismatched conditions, as shown in Fig. 7. Therefore, mismatching filtering and local density control of feature points are required. In this system, homograph matrix[13] is used for mismatching filtering. First, the image coordinates of point P_R on the right image are mapped to the point P_R' on the left image through the homograph matrix, then the distance between P_R' and P_L is calculated, and finally the error limit is adjusted by the set threshold homo_threshold. In addition, set another threshold for the near threshold to adjust the density, which is achieved by detecting the new matching point which has no matching feature points in the threshold range.

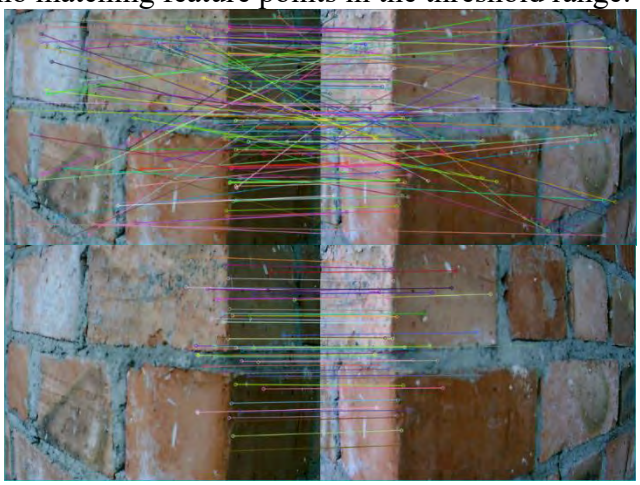


Fig. 7 Comparison before and after mismatching filtering and local density control

4.3 Binocular Vision Subsystem

As for the binocular vision subsystem, the calculation principle of 3D reconstruction has been introduced in the previous part, and it will not be described here. In addition, the optical axis spacing between the left and right cameras in the binocular vision system is 100mm. In order to compare the detection effects of different combinations of corner detection algorithms (FAST, SIFT, SURF, HARRIS) and feature descriptors (SURF, SIFT) in the system, different algorithms were loaded into the system to test the same 50 pairs of images under the same mismatching filtering and density control conditions. The final average results as shown in Table 2:

Table 2 Comparison of algorithm performance

Detection algorithm	FAST	SIFT	SURF	HARRIS	SURF
Feature descriptor	SIFT	SIFT	SIFT	SIFT	SURF
System time (s)	6.996	2.337	16.254	0.790	0.986
Number of matching	106.83	33.57	62.64	10.17	53.83
Number of mismatches	1.17	1.03	1.60	1.33	1.81
Mismatching rate (%)	1.10%	3.07%	2.55%	13.08%	3.36%

It can be seen from Table 2 that the combination of FAST corner detection and SIFT feature descriptor presents the best performance in mismatching rate, but for this system, the detection time is too long to meet the requirements of completing target positioning calculation and action within 8s. Therefore, the combination of SURF corner detection, feature descriptor and FLANN matching

algorithm was finally adopted. After detection, matching, filtering and reconstruction, the results as shown in Fig. 8:

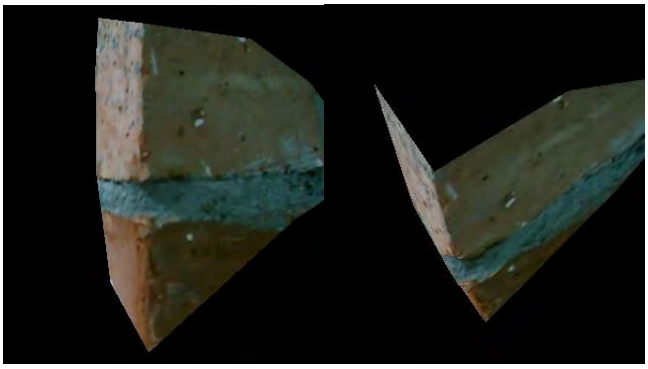


Fig. 8 Results of 3D reconstruction

4.4 Training Samples and Neural Networks

For the acquisition of the base image set, first, divide the 80mm square into a 2mm grid, as shown in Fig. 9, and then each grid point is collected once per degree in the ±10° range based on the square diagonal. A total of 35,301 pairs of images were collected. The calibration of the base image set, each image is marked with a triple (x, y, r), which represents the collection position of the image.

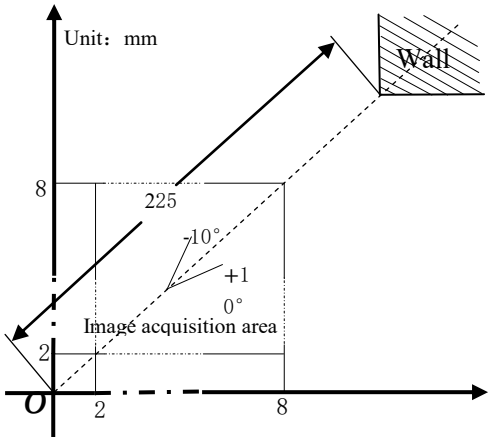


Fig. 9 Schematic diagram of basic image sets collection method

For the generation of training samples, the basic image set is firstly reconstructed in 3D to obtain 3D point cloud information, and then the depth information is preprocessed, and finally the calibration information is merged into a training data. Depth information preprocessing is mainly to extract training data from 3D point cloud information in the same data format.

With the training samples, the neural network can be constructed, learning rate, loss function and training neural network can be defined. A four-layer neural network model is constructed in this system. The number of data nodes in each layer is: 75, 50, 15, 3, the learning rate is $\eta = 0.001$, the activation function is $softmax(x)$、$sigmoid(x)$, and the loss function is the mean square error of the predicted result and the real value.

$$loss = \frac{1}{n}\sum_{1}^{n}(p_i - t_i)^2$$

4.5 Robotic Arm Server and Stereo Display

Because the high-precision motion control ability of the industrial robotic arm is very suitable for image calibration, the image calibration server written by the RAPID language is applied to provide the calibration information for the image acquisition program. As shown in f Fig. 10, which is the calibration result of a point.

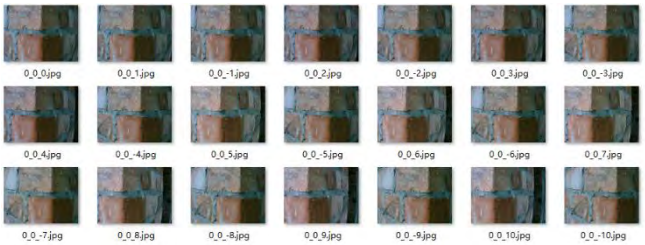


Fig. 10 The calibration result of a point

The stereoscopic display part is for visually displaying the result of the 3D reconstruction on the c ontrol panel of the intelligent plastering robot platform, as shown in Fig. 8, providing the manipulato r with posture information of the arm. It mainly includes triangulation and texture mapping. Texture mapping is implemented based on OpenGL, Triangulation uses Delaunay triangulation[14], as shown in Fig. 11.

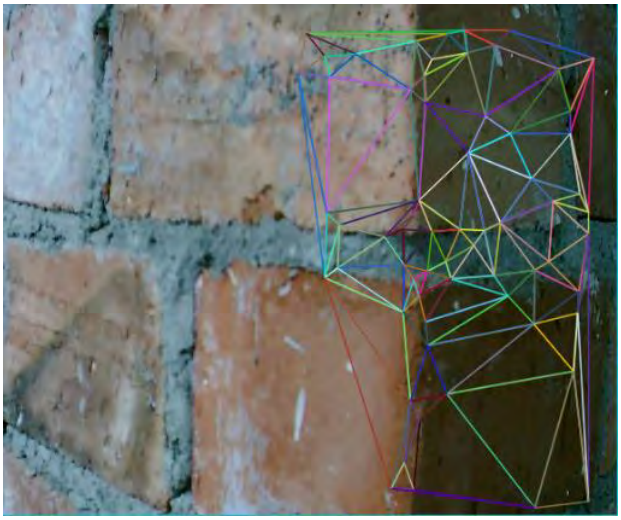


Fig. 11 Triangulation results

4.6 Integration Test Result Analysis

The system is integrated on the intelligent plastering robot platform. Since the control accuracy error of the robot arm is 0.1 mm, the posture information of the target position is recorded using the robotic arm, and then compared with the posture after the pose adjustment. After 80 tests, the adjustment times and final error (absolute error) of each experiment are shown in Table 3. To from the table can be found that 92.5% of posture adjustment can be done within 1 to 3 times, the average single adjustment time is 1.782 s, the average displacement error is ±0.649 mm, the average rotation error of ±0.474 °, reached complete posture positioning system design requirements within 8 s. In addition, figure 12 shows the rendering under the actual construction conditions.

Table 3 Integration test results

Number of adjust	Number of test	Average time	Mean displacement error	Mean rotation error
1	9	1.631s	0.812mm	0.516°
2	21	1.775s	0.637mm	0.529°
3	44	1.803s	0.625mm	0.437°
4	6	1.874s	0.631mm	0.495°
The overall average	*	1.782s	±0.649mm	±0.474°

Fig. 12 Actual plastering results

5. Conclusion

This paper designs a pose adjustment system of robotic arm using binocular vision and machine learning. It has been tested and successfully applied to the intelligent plastering robot platform to meet the design performance requirements. The system performs feature point extraction, matching, filtering, three-dimensional coordinate calculation, density control, pre-processing, regularization point cloud information on the acquired image through binocular vision, and then the posture adjustment information is given by the neural network, and finally by the robot arm performs the actual movement. The experimental results show that the system can meet the positioning requirements of the door, window and frame of the intelligent plastering robot in the process of stability, accuracy and efficiency.

References

[1]. David A Forsyth,Jean Ponce. Computer Vision: A Modern Approach[M]. Beijing: Tsinghua University Press, 2004.

[2]. Bay H, Ess A, Tuytelaars T, Tinne. Speeded-Up Robust Features (SURF)[J]. Computer Vision and Image Understanding. 2008, vol. 110(3), pp. 346-359.

[3]. Xue Lin, Wang Yuliang, Wang Wei, et al. Door detection based on binocular visual in indoor environment [J]. Computer Integrated Manufacturing Systems. 2018. vol. 24(3), pp. 679-688.

[4]. D. G. Lowe, “Object Recognition from Local Scale-Invariant Features” In Proc. of 7th International Conference on Computer Vision. 1999, pp. 1150–1157.

[5]. D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints[C]. International Journal of Computer Vision, 60, vol. 2(2004), pp. 91-110.

[6]. Edward Rosten, Reid Porter, and Tom Drummond, “Faster and better: a machine learning approach to corner detection” in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, vol. 32, pp. 105-119.

[7]. Lindeberg T. Scale-space theory: A basic tool for analyzing structures at different scales[J]. Journal of Applied Statistics, 1994, vol. 21, pp. 225–270.

[8]. Zhang Yuli, Dai Lin, Zhou Weina, et al. A divide and conquer calculation approach of integral graph[J]. Computer Engineering, 2012, vol. 38(22), pp. 232-235.

[9]. Liang Tianxin, Yang Xiaoping, Wang Liang, et al. Review on research and development of memory neural networks.[J]. Journal of Software, vol. 11, pp. 2905-2924.

[10]. Sordoni A, Galley M, Auli M, Brockett C, Ji YF, Mitchell M, Nie JY, Gao JF, Dolan B. A neural network approach to context sensitive generation of conversational responses．In Proc. of the NAACL. 2015. [doi:10.3115/v1/N15-1020]

[11]. Lakkaraju H, Socher R, Manning C． Aspect specific sentiment analysis using hierarchical deep learning．In Proc. of the NIPS Workshop on Deep Learning and Representation Learning. 2014.

[12]. Zhang Zhengyou. A flexible new technique for camera calibration[J]． IEEE Transactions on Pattern Analysis and Machine Intelligence，2000，vol. 22(11), pp. 1330-1334.

[13]. Liu Tingting. Method of eliminate SITF mismatching points based on homography[J]. Journal of Harbin University of Commerce(Natural Sciences Edition), 2016, vol. 32(1), pp. 95-98, 106.

[14]. Yan Zigeng, Jiang Jianguo, GuoDan. [J]. Image matching based on SURF feature and Delaunay triangular meshes[J]. ACTA Automatica Sinica. 2014, vol. 40(6), pp. 1216-1222.