Master of Science

A study on the 3D dimension measurement using robot hand-eye calibration

The Graduate School
of the University of Ulsan
Department of Mechanical Engineering
Chen Man

A study on the 3D dimension measurement using robot hand-eye calibration

Advisor: Prof. Byung-Ryong Lee

A Master's Thesis

Submitted to the Office of Graduate School of
University of Ulsan
in partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Chen Man

School of Mechanical Engineering

Ulsan, Republic of Korea

December 2019

CHEN MAN 의

공학석사 학위 논문을 인준함

심사위원	안경관	인
심사위원	양순용	인
심사위원	이병룡	인

울산대학교 대학원

기계자동차공학과

2019 년 12 월

A study on the 3D dimension measurement using robot hand-eye calibration

This certifies that the master's thesis

of Man Chen is approved by

Kyong Kwan

Committee Chair Prof. Kyung-Kwan Ahn

Committee Member Prof. Soon-Yong Yang

Lee Byung Ryong

Committee Member Prof. Byung-Ryong Lee

School of Mechanical Engineering

Ulsan, Republic of Korea

December 2019

ABSTRACT

A study on the 3D dimension measurement using robot hand-eye calibration

Man Chen

Chair of Advisory Committee: Prof. Kyung-Kwan Ahn

December 2019

With the continuous progress of science and technology, industrial robots are entering factories to replace human beings. Industrial robots are highly efficient, but they mainly detect and locate fixed targets on a single production line in a specific environment. For this reason, the vision sensor and industrial robot are combined to complete the fixed target work by visual guidance. This paper mainly studies the techniques of camera calibration, feature point extraction and three-dimensional positioning in the binocular stereo vision system. A visual positioning and ranging system suitable for industrial production is constructed, and the distance measurement between multiple target points is realized. The results show that the average error of three-dimensional positioning ranging is less than 0.5 mm, which proves the feasibility of the method. At the same time, the hand-eye matrix is calculated by matrix direct product, and the target point pose of the world coordinate system is converted into the pose of the robot coordinate system, which provides a basis for industrial production.

The main work is as follows:

(1) Build a simple hardware platform with a robotic arm, binocular camera and computer.

(2) The theoretical basis and principle of camera calibration are studied.Based on the linear calibration method of perspective transformation model in MATLAB software environment obtain camera parameters and distortion parameters.

(3) The two images are acquired by using a binocular vision camera. The pre-processing of the acquired image is completed, and the specified target point is searched in the image by template matching to obtain the two-dimensional image parallax.

(4) Based on the internal and external parameters of the camera and matching points, the depth of the feature points is calculated according to the stereo vision measurement principle, and corresponding three-dimensional space information is obtained.

(5) The hand-eye transformation matrix of the robot is calculated by using the direct product of the matrix and the eigenvector. The spatial relationship between the camera coordinate system and the robot arm coordinate system is determined by hand-eye calibration, and the relative position of the target point relative to the robot end tool is calculated, which provides a data basis for the spatial operation.

Keywords : Camera calibration，binocular vision , hand-eye calibration

DEDICATION

For my beloved family

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my academic advisor, Prof. Byung-Ryong Lee, for the continuous support of my research over the past two years. His excellent guidance, caring, patience provided me in all the time of research and writing of this thesis. I have learned many things since I became Dr. Lee’s students. I greatly appreciate him as my advisor. Without his passionate participate and input, I could not have imagined finishing my research within two years successfully.

During the period of two years, many friends are helpful to color my life. I would also like to thank all the members of Laboratory for their assistances in many aspects that I cannot list them all due to limited spaces. It has been a wonderful experience for me to work and play with them, and they helped me in numerous ways during two years.

Last but not least, I owe more than thanks to my whole family members. Especially, I want to thank parents for their support and love they gave me during the passing years. Without their support, it’s very hard for me to finish my graduate education.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Research Background and Significance

The development of robots stems from the urgent needs of people. Invented industrial robots in order to liberate from dangerous work or repeated labor, and invented service robots for a better life [1]. The vigorous development of science and technology makes it possible to create robots of various types and different scenarios, which are widely used in industry, agriculture, aerospace and other fields.

Manipulators are widely used in various fields because of their high stability, high efficiency, low environmental requirements and large storage space and fast operation speed. Manipulators can work in harsh environments where human beings are not suitable or unable to work, or can replace human beings for repetitive operations, thus reducing the risk and burden of human work and improving production efficiency. However, the manipulator has the disadvantages of low flexibility and low degree of autonomy. Traditional manipulators can only grasp the target according to the planned path, and can’t obtain external information. Therefore, in unknown or changing environment, it is necessary to changes, otherwise it is difficult to complete the grasping task. In order to enhance the flexibility of the manipulator control system and make it have the ability to perceive the surrounding environment, machine vision has been widely used in various robotic and has become an important research direction in the field of robotics [2,3].

The combination of robotic arm and machine vision can improve the perception ability of surrounding environment, intelligent decision-making and behavior ability of robotic arm, and further promote the intelligent development of robotic arm. Compared with monocular vision, binocular stereo vision technology can flexibly acquire the three-dimensional information of the target, which has obvious advantages. At present, binocular vision has been used more and more widely in many fields such as industrial inspection, robot navigation, medical machine imaging, pose detection, and military aviation [4].

1.2 Research Content and Chapter Arrangement

This paper mainly studies the application of vision positioning system in industrial production, and establishes a vision-based robot target positioning platform. The proposed binocular stereo vision system consisted of a PC, two CCD colour cameras, a calibration board and software system. The cameras (model TGCAM-2000S) produced by TG SanBao company in Korea had a digital video output of 2000 effective pixels, and they were placed in parallel and mounted on the manipulator that had six DOF in order to obtain images flexibly. The Image acquisition was programmed in OpenCV 3.0 (supplied by Intel Corporation at Santa Clara in California, USA), cameras calibration module was programmed in Matlab2018a (supplied by MathWorks Corporation at Nedik in Massachusetts, USA) ran in the PC with Windows 7 operating system.

This paper mainly studies the following four aspects:

Chapter 1: Introduction. It mainly expounds the research background and significance.

Chapter 2: Camera Calibration. It mainly introduces the basic theory and principle of camera calibration, and uses Zhang Zhengyou's chessboard method to calculate the internal and external parameters and distortion parameters of the camera.

Chapter 3: Stereoscopic vision. The target is captured using a binocular camera and the pixel coordinates of the target point are calculated. The depth of the target is calculated by the parallax method, and finally the three-dimensional coordinate value of the target is obtained.

Chapter 4: Hand-eye calibration. The theory and principle of calculating the hand-eye matrix using Rodrigue vector rotation formula and matrix direct product are introduced respectively. The hand-eye matrix is calculated by the matrix direct product method to determine the spatial relationship between the camera coordinate system and the robot arm coordinate system, and the relative position of the target point relative to the robot terminal tool is calculated.

Chapter 5: Conclusions. Summarize the work of this paper and analyze the shortcomings of this paper.

CHAPTER II

CAMERA IMAGING MODEL AND CALIBRATION

2.1. Camera Model

Camera imaging mainly analyses how to associate the points in three-dimensional space with their projection points in the camera imaging plane. The camera imaging plane is actually composed of pixel, and the generated image is based on the pixel. Therefore, the camera imaging model needs the conversion relationship between the world coordinate system and the pixel coordinate system. In order to calculate the conversion relationship between the world coordinate system and the pixel coordinate system, it is necessary to add the camera coordinate system and the image coordinate system as the intermediate conversion coordinate system. The conversion relationship between the coordinate systems is shown in Figure 2.1. In this paper, we will analyze the conversion relations between pixel coordinate system, image coordinate system, camera coordinate system and world coordinate system.

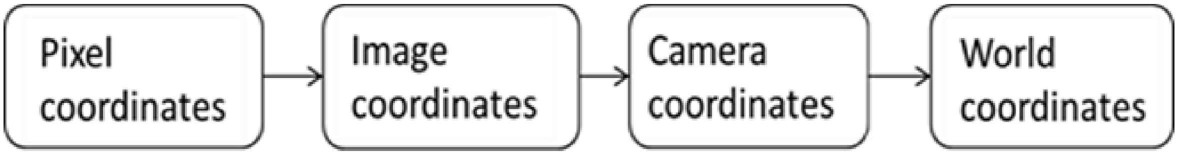


Fig.2.1 Coordinate System Conversion

World coordinate system: User-defined coordinate system of three-dimensional world, describing the location of the target object in the real world.

Camera coordinate system: The coordinate system established on the camera, which is defined to describe the position of objects from the perspective of the camera, serves as the middle link between the world coordinate system and the image/pixel coordinate system.

Image coordinate system: Describe the projection and transmission relationship between the object in the imaging process from the camera coordinate system to the image coordinate system, so as to further obtain the coordinates in the pixel coordinate system.

Pixel coordinate system: The coordinate describing the image point on the digital image after imaging is the coordinate system where the information we really read from the camera is located.

There is an imaging process when capturing a target image with a camera. At present, commonly used imaging models mainly include linear imaging models and nonlinear imaging models [5]. In this paper, a simplified pinhole camera model is used instead of a lens model to analyze the camera imaging process. The pinhole model, while briefly describing the imaging process of the camera, is very accurate. The geometric relationship of the pinhole model is shown in Fig 2.2.

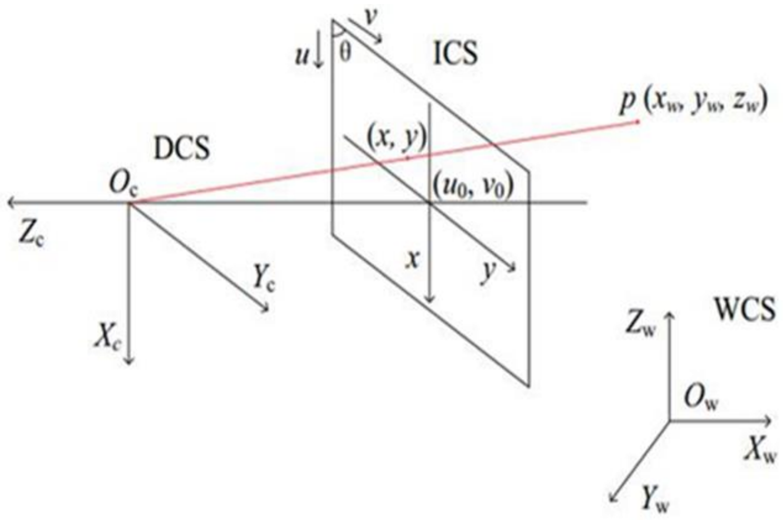


Fig.2.2 Imaging Model of Pinhole Camera

Generally speaking, the calibration process is divided into two parts:

The first step is to transform the world coordinate system to the camera coordinate system. This step is the transformation from three-dimensional point to three-dimensional point. Including R, t (camera external parameters, determine the position and orientation of the camera in a three-dimensional space) and other parameters.

The second part is the transformation from camera coordinate system to pixel coordinate system. This step is the transformation from three-dimensional point to two-dimensional point. Including K (camera internal parameters, is the approximation of the physical characteristics of the camera) and other parameters.

2.1.1. World coordinate system

The transformation between camera coordinate system and world coordinate system is the transformation between two three-dimensional coordinate systems [6]. The principle is the same as the coordinate transformation of the manipulator. The transformation can be completed by determining the position and attitude relationship between the coordinate systems. The relationship between the camera coordinate system (DCS) and the world coordinate system (WCS) is as follows:

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{2.1}$$

The camera coordinate system is located inside the lens, so the position of the origin and the direction of each coordinate axis can not be accurately known from the outside. Therefore, it is necessary to use camera calibration to determine the parameters$R_1{}^T, R_2{}^T, R_3{}^T, T_1, T_2, T_3$. These six parameters are called camera external parameters. Where R is a rotation matrix and T is a translation matrix, which describes the transformation relationship between the camera coordinate system and the world coordinate system [7].

2.1.2 Camera coordinate system

The camera imaging geometry is shown in the Figure 2.3. Where O is the camera's optical center, X_C and Y_C axes are parallel to the X and Y axes of the imaging plane coordinate system. The Z_C axis is the optical axis of the camera and is perpendicular to the image plane. The intersection of the optical axis and the image plane is the image origin o1. A Cartesian coordinate system composed of points O and X_C, Y_C, and Z_C axes is called a camera coordinate system. OO_1 is the camera focal length.

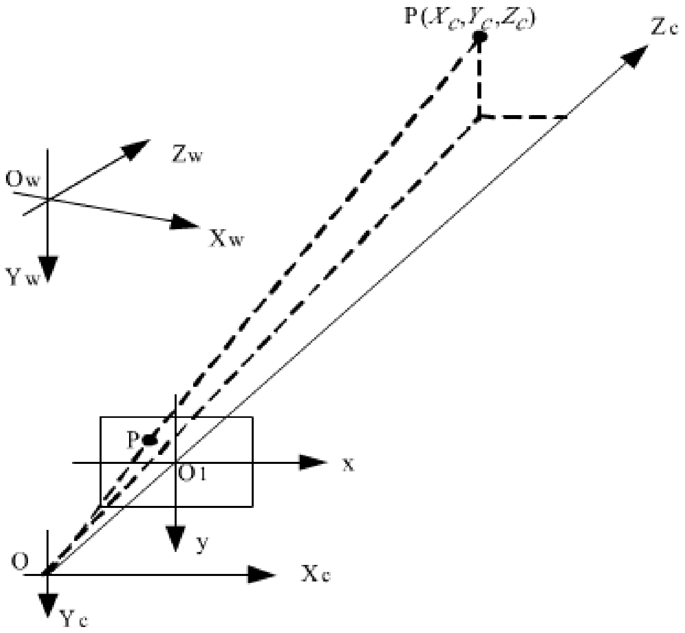


Fig. 2.3 Camera Coordinate System and World Coordinate System

The image plane is perpendicular to the Z_c axis of the optical axis, and the distance from the projection center is f (f is the focal length of the camera). According to the trigonometric proportional relation, it can be concluded that:

$$\frac{x}{f} = \frac{X_c}{Z_c} \tag{2.2}$$

$$\frac{y}{f} = \frac{Y_c}{Z_c} \tag{2.3}$$

So:

$$\mathrm{x} = \frac{f * X_c}{Z_c} \tag{2.4}$$

The above process of mapping point P with coordinates (x, y, z) to point p with coordinates (x, y) on the projection plane is called projection transformation.

The projection transformation matrix is:

$$\mathrm{Z}_c\begin{bmatrix}x\\y\\1\end{bmatrix}=\begin{bmatrix}f&0&0\\0&f&0\\0&0&1\end{bmatrix}\begin{bmatrix}X_c\\Y_c\\Z_c\end{bmatrix} \tag{2.5}$$

2.1.3 Image coordinate system

The coordinate system is established with the upper left corner or the lower left corner of the image plane as the origin. As shown in Figure 2.4，Suppose that the origin O_0 of the plane coordinate system is located in the upper left corner of the image, u-axis horizontally to the right and v-axis vertically. The coordinate system is established by taking the intersection point O_1 of image plane and optical axis as the origin point. The horizontal axis is x axis and the vertical axis is y axis.

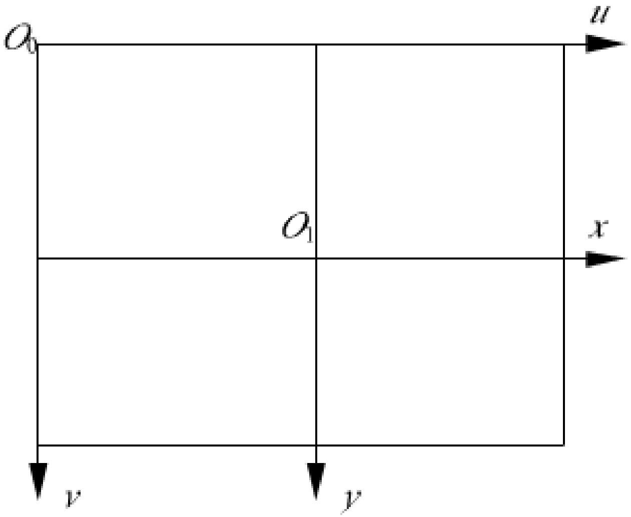


Fig.2.4 Image Coordinate System and Pixel Coordinate System

The image coordinate system and the pixel coordinate system are on the same plane, but the origin is not the same. (u,v) represents coordinates in units of pixels, and (x,y) represents coordinates of physical units. Assuming that the physical size of each pixel is dx*dy, there are the following relationships:

$$\mathrm{u} = \frac{x}{d_x} + u_0 \tag{2.6}$$

$$\mathrm{v} = \frac{y}{d_y} + v_0 \tag{2.7}$$

The form of matrix is as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{d_x} & 0 & u_0 \\ 0 & \frac{1}{d_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.8}$$

So:

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{d_x} & 0 & u_0 \\ 0 & \frac{1}{d_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & k_s & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} *$$

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = K * R * \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{2.9}$$

In the formula, K represents the internal parameter matrix of the camera, $[u \quad v \quad 1]$represents the homogeneous coordinates of the object projected onto the image plane, $[r_1 \quad r_2 \quad r_3\ ]$and t are the external parameters of the camera, representing the rotation matrix and translation vector of the camera-centered coordinate system relative to the world coordinate system, respectively. The position of multiple corners can be obtained by checkerboard lattice, and the camera's internal and external parameters can be obtained by solving equations.

2.2. Distortion

In the actual imaging process, nonlinear distortion is generally present in consideration of lens distortion. The distortion is mainly divided into radial distortion and tangential distortion.

2.2.1 Radial distortion

Radial distortion refers to a given image point moving inward or outward from its ideal position. That is, Pillow radial distortion and bucket radial distortion. As shown in Figure 2.5. It is mainly caused by defects in the change of radial curvature of lens surface. When light passes through the lens, the light that is farther away from the optical axis bends much more than the light near the optical axis [8].The negative radial displacement of image point is called barrel distortion, and the positive radial displacement is called pillow distortion. Radial distortion is the main reason that affects the accuracy of industrial machines. When the image is distorted, the image points are not in the position described by the linear model, which will lead to the inaccurate correspondence of the points in stereo matching and the distortion of the object after three-dimensional reconstruction.

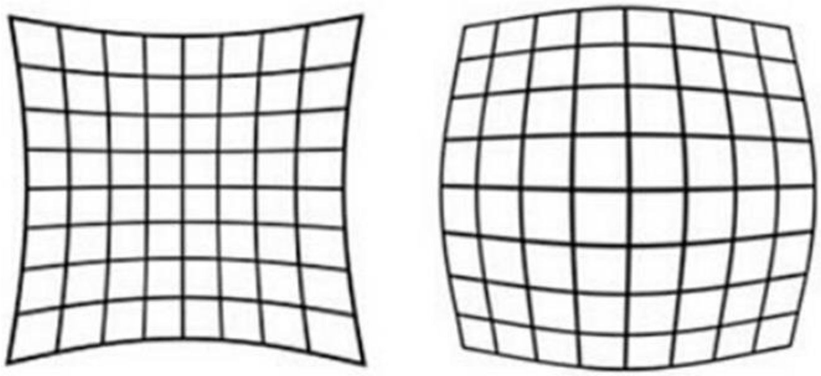


Fig.2.5 Pillow radial distortion (left) and Bucket radial distortion (right)

The distortion of the optical axis center of the imager is 0. It moves to the edge along the radius of the lens, and the distortion becomes more and more serious. The mathematical model of distortion can be described by the first several terms of Taylor series expansion around the main point. According to the position of a point in the radial direction, the adjustment formula is as follows:

$$x_1 = x(1 + k_1 * r^2 + k_2 * r^4) \quad (2.10)$$

$$y_1 = y(1 + k_1 * r^2 + k_2 * r^4) \quad (2.11)$$

Here (x, y) is the original position of the distortion point on the imager, (x_1, y_1) is the position after the distortion correction.

2.2.2 Tangential distortion

Tangential distortion is caused by the lens itself and the camera plane or image plane is not parallel, this situation is mostly caused by installation deviation. As shown in Figure 2.6.

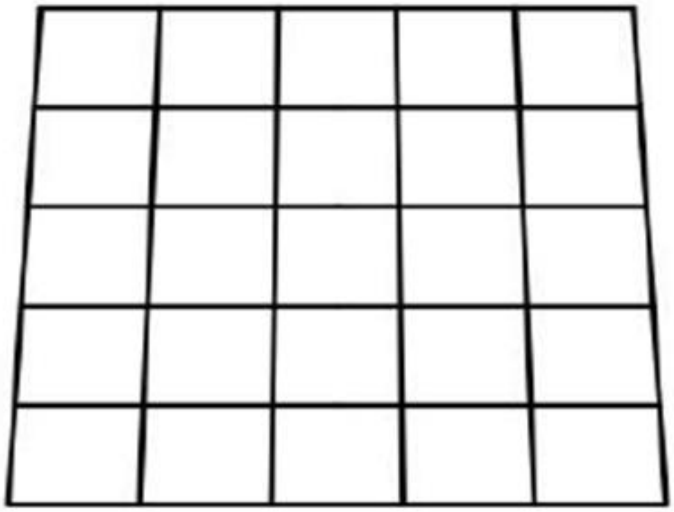


Fig.2.6 Tangential distortion

The distortion model can be described by two additional parameters P1 and P2:

$$x_2 = x + [2 * p_1 * y + p_2 * (r^2 + 2 * x^2)] \tag{2.12}$$

$$y_2 = y + [2 * p_2 * x + p_1 * (r^2 + 2 * y^2)] \tag{2.13}$$

Here (x, y) is the original position of the distortion point on the imager, $(x_{2,} y_2)$ is the position after the distortion correction.

2.3. Camera calibration

In the process of image measurement and the application of machine vision, in order to determine the relationship between the three-dimensional geometric position of a point on the surface of a space object and its corresponding points in the image, the geometric model of camera imaging must be established. These geometric model parameters are camera parameters. These parameters must be obtained by experiment and calculation. The process of solving these parameters is called camera calibration. Whether in image measurement or machine vision applications, the calibration of camera parameters is a key link. The accuracy of the calibration results and the stability of the algorithm directly affect the accuracy of the results of camera work.

2.3.1 Camera calibration method

Camera calibration methods include traditional camera calibration method, active vision camera calibration method and camera self-calibration method.

The traditional camera calibration method needs to use the calibrator with known size. By establishing the correspondence between the points with known coordinates on the calibrator and their image points, the internal and external parameters of the camera model can be obtained by using certain algorithms. According to the different calibration objects, they can be divided into three-dimensional calibration objects and plane calibration objects. Three-dimensional calibration objects can be calibrated by a single image with high calibration accuracy, but it is difficult to process and maintain high-precision three-dimensional calibration objects. Planar calibrator is simpler and more accurate than three-dimensional calibrator, but two or more images must be used in calibration. The traditional camera calibration method always needs calibration objects in the calibration process, and the accuracy of calibration objects will affect the calibration results. At the same time, the inappropriate placement of calibrators in some occasions limits the application of traditional camera calibration methods.

At present, the self-calibration algorithm mainly uses the constraints of camera motion. Camera motion constraints are too strong, so it is not practical in practice. The use of scene constraints is mainly based on some parallel or orthogonal information in the scene. The intersection of spatial parallel lines on the plane of camera image is called vanishing point, which is a very important feature in projective geometry. So many scholars have studied the camera self-calibration method based on vanishing point. The self-calibration method is flexible and can calibrate the camera online. However, because it is based on absolute quadratic curve or surface, its robustness is poor.

The camera calibration method based on active vision refers to the calibration of the camera with certain motion information of the camera. This method does not need calibrator, but needs to control the camera to do some special motion. Using the particularity of this motion, the camera internal parameters can be calculated. The advantage of the camera calibration method based on active vision is that the algorithm is simple and the linear solution can be obtained, so the robustness is high. The disadvantage is that the cost of the system is high, the experimental equipment is expensive, the experimental conditions are high, and it is not suitable for situations where the motion parameters are unknown or uncontrollable.

In 1999, Zhang Zhengyou proposed a camera calibration method based on moving plane template. Zhang's calibration method is between the traditional calibration method and the self-calibration method, and is based on the two-dimensional calibration board for calibration [9]. Fixed the camera to a certain position, the mobile camera takes multiple pictures from different directions, and calculates camera parameters by calibrating the corresponding relationship between each feature point and its plane image points. In this model, a scene view is formed by projecting 3D points into the image plane using a perspective transformation [10].

2.3.2 Zhang's chessboard calibration method

In this paper, Zhang's calibration method is used to analyze and study the calibration process and principle of the calibration method. Suppose that the world coordinate system is constructed on the plane Z = 0. Then the homography is calculated. Let Z = 0 convert the upper form to the following form:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & 0 & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{21} & t_1 \\ r_{12} & r_{22} & t_2 \\ r_{13} & r_{23} & t_3 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} = K * [r_1 \quad r_2 \quad t] * \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \qquad (2.14)$$

Here we introduce a homology matrix H with a size of 3 x 3. H is written in the form of column vectors $\mathrm{H} = [h_1, h_2, h_3]$, each h being a 3 *1 vector. The homography matrix H is composed of physical transformation (rotation, translation) and camera internal parameters [11].

$$\mathrm{H} = [h_1 \quad h_2 \quad h_3] = K * [r_1 \quad r_2 \quad t] \qquad (2.15)$$

So:

$$r_1 = K^{-1} * h_1 \tag{2.16}$$

$$r_2 = K^{-1} * h_2 \tag{2.17}$$

We know R= [r1, r2, r3], because Z = 0, so r3 = 0. R is an orthogonal matrix. That is, the transformation of R equals the inverse of R. Each column vector of an orthogonal matrix is orthogonal and unitary (modulus=1), so r1 and r2 are orthogonal to each other. Orthogonal means two things: the dot product of two vectors is 0, and the length of two vectors is equal. We use these two constraints to solve the problem.

$${r_1}^T * r_2 = 0 \tag{2.18}$$

$$\| r_1 \| = \| r_2 \| = 1 \tag{2.19}$$

It can be obtained:

$${h_1}^T * K^{-T} * K^{-1} * h_2 = 0 \tag{2.20}$$

$${h_1}^T * K^{-T} * K^{-1} * h_1 = {h_2}^T * K^{-T} * K^{-1} * h_2 \tag{2.21}$$

Thus,

$${h_1}^T * B * h_2 = 0 \tag{2.22}$$

$${h_1}^T * B * h_1 = {h_2}^T * B * h_2 \tag{2.23}$$

$$\mathrm{B} = K^{-T} * K^{-1} = \begin{bmatrix} B_{11} & B_{21} & B_{31} \\ B_{12} & B_{22} & B_{32} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} = \begin{bmatrix} k_x & 0 & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}^{-T} * \begin{bmatrix} k_x & 0 & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} =$$

$$\begin{bmatrix} \frac{1}{{k_x}^2} & 0 & \frac{-u_0}{{k_x}^2} \\ 0 & \frac{1}{{k_y}^2} & \frac{-v_0}{{k_y}^2} \\ \frac{-u_0}{{k_x}^2} & \frac{-v_0}{k_y} & \frac{{u_0}^2}{{k_x}^2} + \frac{{v_0}^2}{{k_y}^2} + 1 \end{bmatrix} \tag{2.24}$$

Because B is a symmetric matrix, there are only six effective elements. The general form is expanded as follows:

$$\mathrm{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T \tag{2.25}$$

Thus,

$$h_1{}^T * B * h_2 = [h_{11} \quad h_{12} \quad h_{13}]^T * \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} * \begin{bmatrix} h_{21} \\ h_{22} \\ h_{23} \end{bmatrix} = \begin{bmatrix} h_{11} * h_{21} \\ h_{12}h_{21} + h_{11}h_{22} \\ h_{13}h_{21} + h_{11}h_{23} \\ h_{12}h_{22} \\ h_{13}h_{22} + h_{12}h_{23} \\ h_{13}h_{23} \end{bmatrix}^T * \begin{bmatrix} B_{11} \\ B_{12} \\ B_{13} \\ B_{22} \\ B_{23} \\ B_{33} \end{bmatrix} \tag{2.26}$$

It can be obtained:

$$V_{12}{}^T * b = 0 \tag{2.27}$$

$$(V_{11}{}^T - V_{22}{}^T) * b = 0 \tag{2.28}$$

There are six unknowns in the formula, so more than three images can calculate all the parameters.

In the above calibration process, the distortion factor of the camera lens was not considered. For systems with high precision requirements, the distortion is not negligible. Therefore, the distortion factor should be considered, and the maximum likelihood estimation algorithm is used to optimize the results, so that the algorithm has better robustness [12].

2.3.3 MATLAB Stereo calibration

The calibration result of MATLAB calibration toolbox is more accurate than that of OpenCV3.0, this paper uses MATLAB to calibrate the camera.

The specific steps of calibration are as follows:

(1) The experiment used a standard 10*7 and each small square was a black and white square of 16.3mm*16.3mm, as shown in Figure 2.7. The checkerboard was printed and attached to a hard and flat plastic calibration plate.

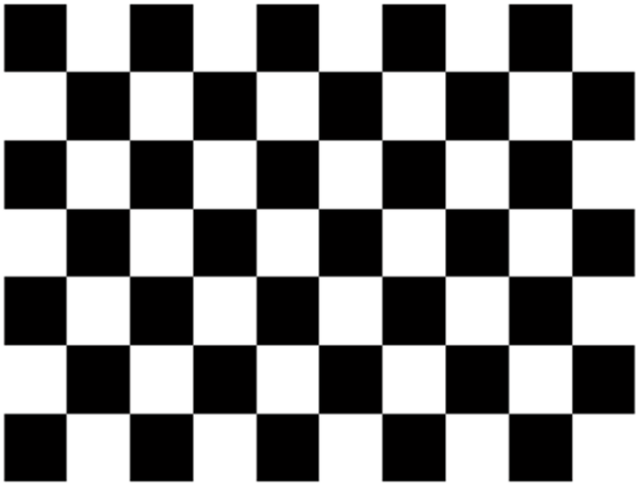


Fig.2.7 Two-dimensional calibration board

(2) Fix the binocular camera to the end of the robot arm, and move the robot arm so that the binocular camera can take pictures of the calibration plate from different positions at the same time. In the acquisition process, the calibration image should be placed in the middle of the camera field of view.

(3) In order to improve the accuracy of the experimental results, a total of 10 pairs of image pairs taken at different angles were collected in this experiment. Read the camera image for stereo calibration.

The calibration image is as follows:

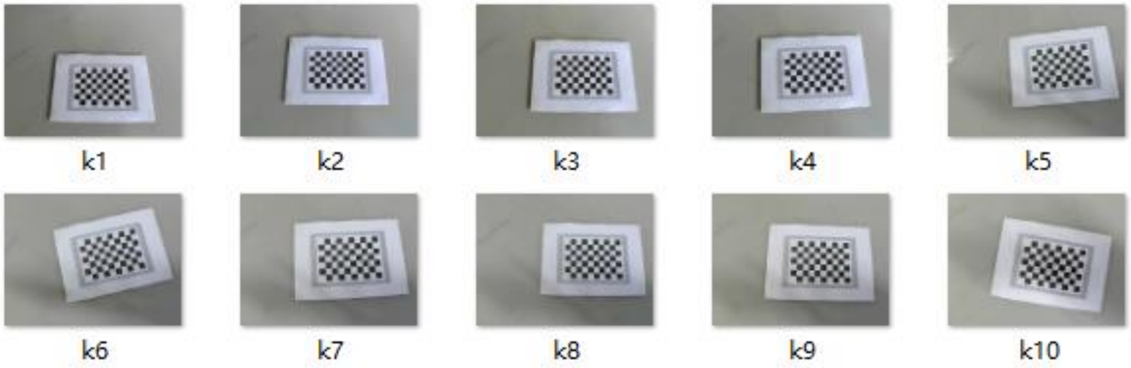


Fig.2.8 Left Camera Calibration Image Set

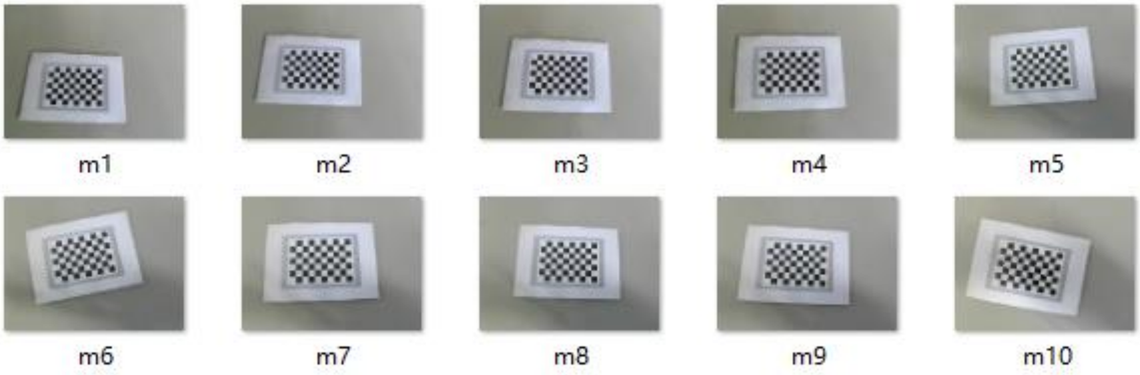


Fig.2. 9 Right Camera Calibration Image Set

Position relationship between chessboard calibration board and two cameras in calibration:

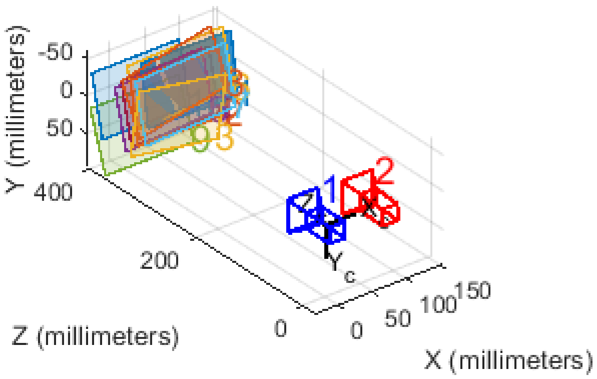


Fig.2.10 Positional relationship between the camera and the calibration plate

After stereo calibration, the results are as follows:

Table 1 Calibration result of stereo camera

Camera	parameter	Result
Left Camera	Focal Length	[233.9621 234.1415]
	Principal Point	[157.6331 116.1187]
	Distortion	[-0.0346,-0.2230,0.7652,-0.0017,-0.0030]
Right Camera	Focal Length	[232.3806 232.9353]
	Principal Point	[164.6956 118.4452]
	Distortion	[-0.0120,-0.4240,1.1344,-0.0003,0.0028]
Binocular camera	Rotation vector	[0.9983 -0.0253 -0.0518 0.0246 0.9996 -0.0138 0.0521 0.0125 0.9986]
	Translation vector	[-76.1284 -3.9527 -1.5264]

2.4. Conclusion

In the binocular vision system, the calibration of the binocular camera has a great influence on the accuracy of the final reconstruction result. Therefore, in this chapter, the basic theories and concepts such as the camera calibration principle, the camera calibration model, and the camera calibration method are studied in detail. Considering the actual needs, the Zhang Zhengyou board method was chosen as the camera calibration method for this experiment. The experimental results show that Zhang's chessboard method has higher accuracy and meets the experimental requirements.

CHAPTER III

BINOCULAR STEREO VISION

Vision is the main source of human access to outside information. In large-scale industrial production processes, the use of artificial vision often fails to meet the requirements of high efficiency and high-precision product quality, and the use of machine vision methods can greatly improve the degree of automated production [13]. Combine the manipulator with machine vision to enhance the ability of the arm to sense the surrounding environment. If the main computer is the brain of the industrial robot and the manipulator is the arm of the industrial robot, then the vision system is the eye of the industrial robot.

Binocular stereo vision technology is based on the stereoscopic imaging principle of the human eye. The object to be measured is taken from different angles by using two cameras of the same performance, and then the three-dimensional coordinates of the target are acquired by techniques such as camera calibration, image calibration, stereo matching， feature extraction, and stereo measurement. The main difficulty of the binocular stereo vision method is stereo matching [14]. Since the center point of the circle is used as a feature point, it can be extracted by image processing technology, so stereo matching can be avoided.

3.1. Image acquisition

Connect the USB line used by the camera with the computer interface, and Image pairs are acquired by OPENV programming. As shown in Figure 3.1.

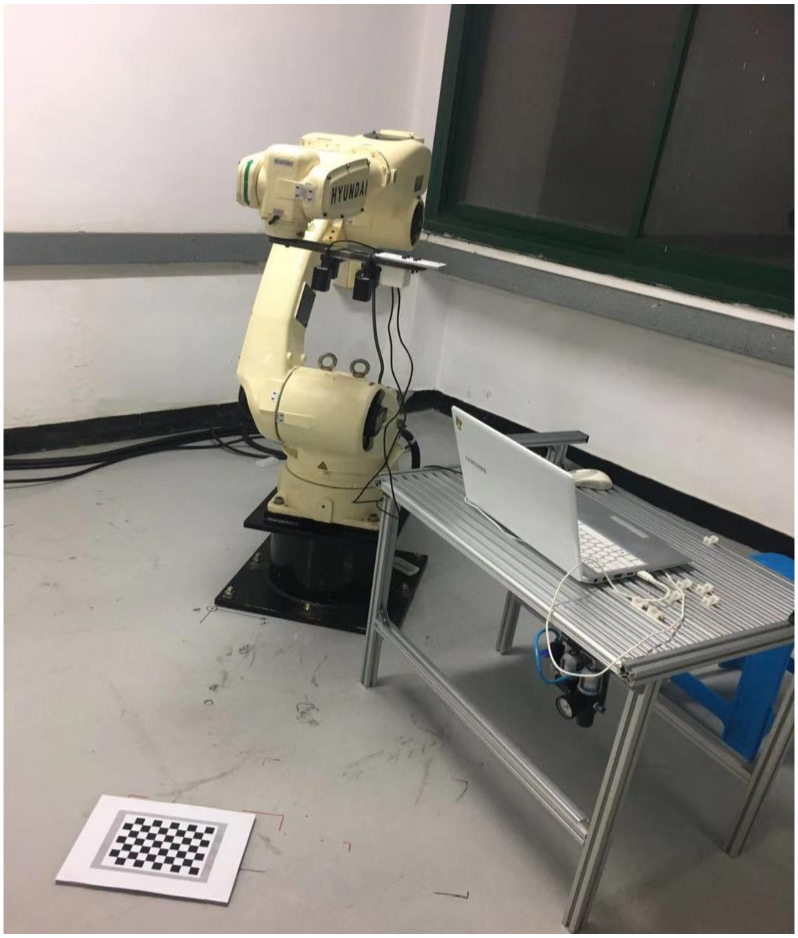


Fig.3.1 Image Acquisition Equipment

3.2. Stereo rectify

There are many errors in the camera imaging process, such as noise, brightness difference, occlusion [15], distortion and the two cameras are not placed in parallel, which will cause errors in the corresponding points of the image pair. So it is necessary to stereo correct the original image. The correction matrix can be calculated from the camera parameters obtained in the previous section. Then, the correction matrix is used to obtain the corrected picture. The specific role of stereo correction is shown in the figure below.

Before stereoscopic correction:

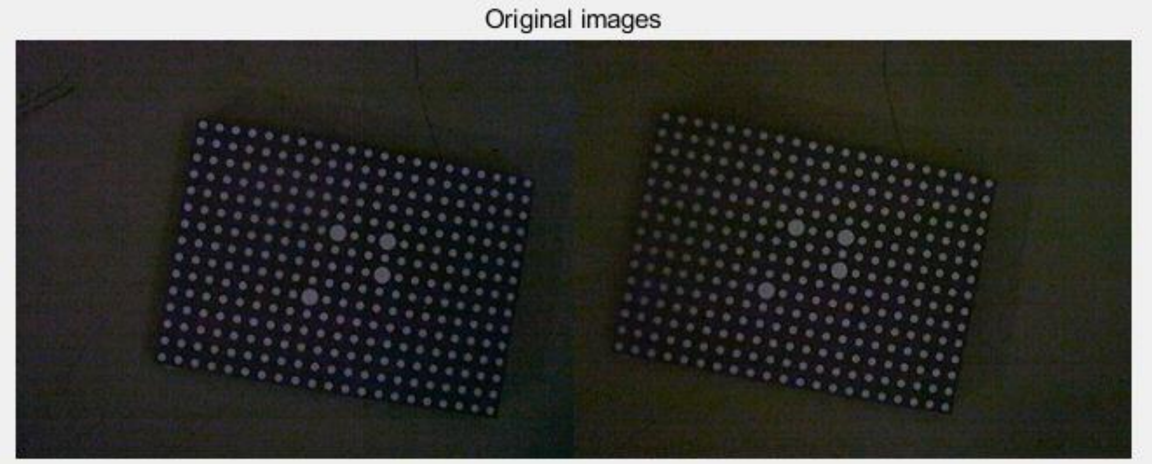


Fig.3.2 Original image

After stereoscopic correction：

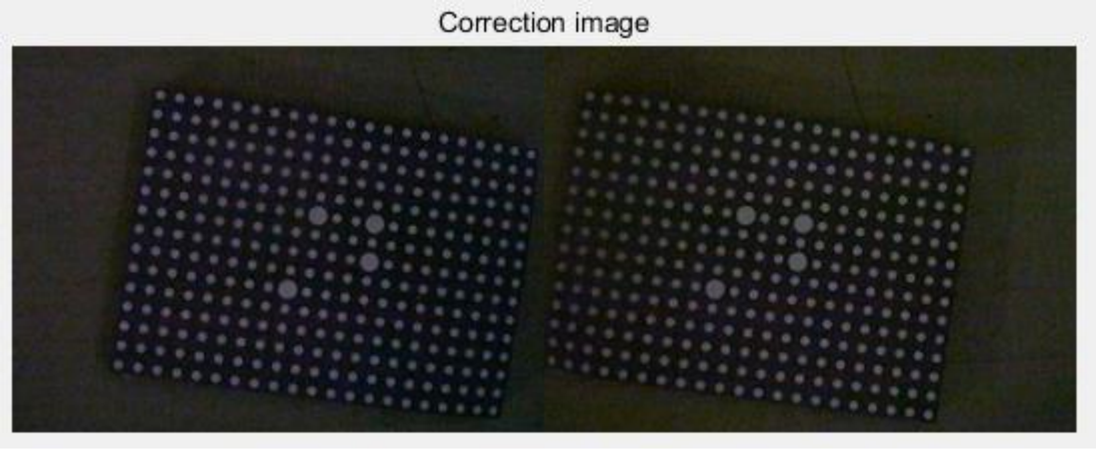


Fig.3.3 Correction image

3.3. Image processing

3.3.1. Color space conversion

The image acquired by the camera is a color image. Color images are mainly divided into two types: RGB and HSI.

The RGB color space is composed of three basic colors of red, green and blue. It can be superimposed into any color by combining the three basic colors in different proportions. The RGB color space can be represented by a spatial cube space model. The three basic colors are located on the unit length of the coordinate axis, and the brightness of each basic color is divided by the maximum value of the brightness to represent the coordinate value of the color, thereby establishing an RGB color space. As shown in Fig. 3.4.

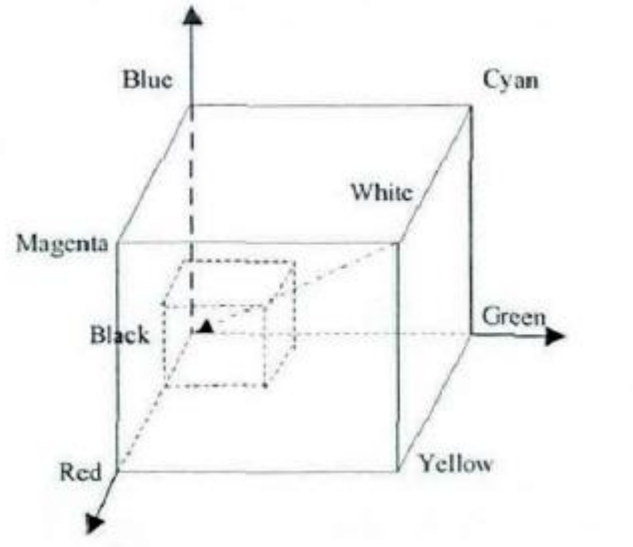


Fig.3.4 RGB color space

The HSI color space is composed of hue, saturation, and intensity. The hue indicates the type of color, the saturation indicates the vividness of the color, and the intensity indicates the degree of color shading. The HSI color space can be represented by a conical space model, as shown in Fig. 3.5. The center of the conical bottom is taken as the coordinate origin, and the hue is represented 360 degrees along the ground with the origin as the center. The distance from the center line passing through the origin indicates saturation, and the distance from the ground indicates intensity.

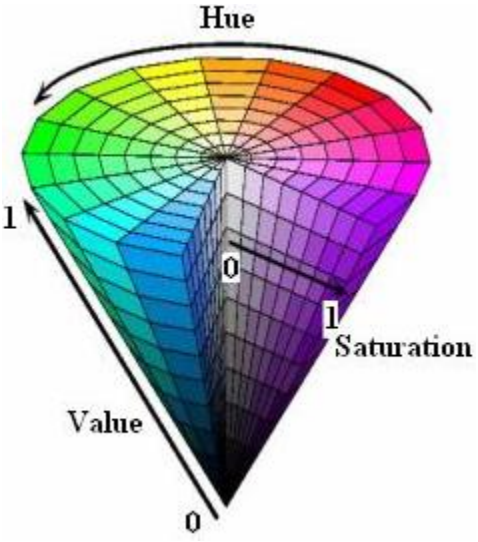


Fig.3.5 HSI color space

The pixels of each image typically correspond to a particular location in a two-dimensional space, and have one or more sample values associated with that point that constitute a value. In order to make the boundary features of the image clearer, the color image is usually converted into a gray image. A grayscale image, also known as a grayscale image, can be represented by an intensity value (intensity) from 0 (black) to 255 (white). Between 0 and 255 represents a different gray level.

The process of converting a color image into a grayscale image is called grayscale processing of the image. Each assumed color in a color image is determined by three components R, G, and B, and each component has 255 values, so that a point can have a range of colors of more than 16 million. On the other hand, in the digital image processing, the image of each format is first converted into a grayscale image so that the calculation amount of the subsequent image becomes less. The description of the grayscale image, like the color image, still reflects the distribution and characteristics of the overall and local chromaticity and brightness levels of the entire image. For the method of graying, there are a component method, a maximum value method, an average value method, a weighted average method, and the like. In this paper, the image is grayscaled using the weighted average method. The expressions it implements are as follows:

$$Gray=R*0.3+G*0.59+B*0.11 \tag{3.1}$$

Call the rgb2gray ( ) function to convert an RGB color space image to a grayscale image. As shown in Fig. 3.6.

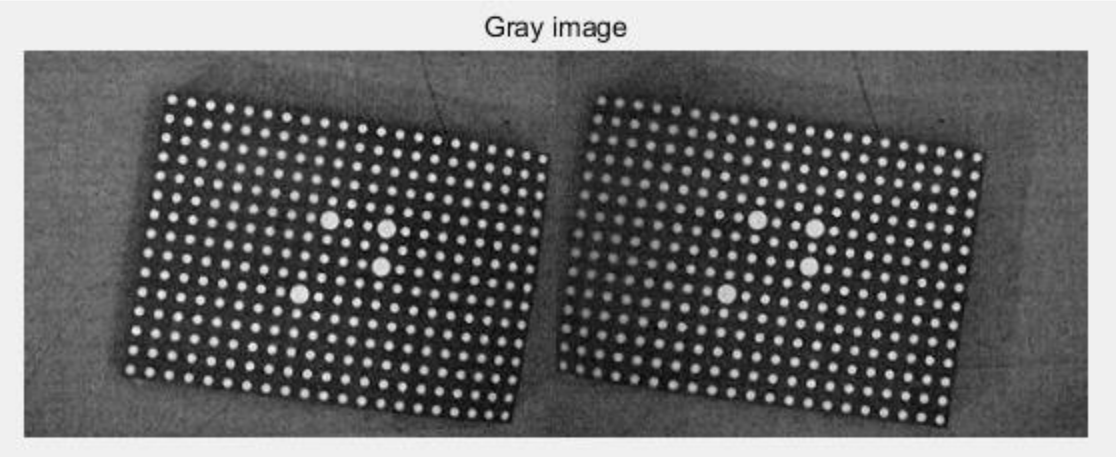


Fig.3.6 Gray image

3.3.2. Image thresholding

The acquired grayscale image is binarized. For binarization, the purpose is to classify the target user background to prepare for the identification of subsequent targets. The most common method for gray image binarization is the threshold method. He uses the difference between the target and the background in the image to set the image to two different levels, and selects an appropriate threshold instead of a replacement for the target or the background. Thereby a binarized image is obtained.

For the threshold method binarization, assuming that the threshold is set to T, the value can be divided into two parts with T as the boundary, The pixel with the gray value greater than or equal to the threshold is set as the maximum gray value, and the pixel with the gray value smaller than the threshold is set to the gray value of 0.

The formula for binarization is as follows:

$$\mathrm{dst}(\mathrm{x},\mathrm{y}) = \begin{cases} max, & src(x,y) \geq thresh \\ 0, & src(x,y) < thresh \end{cases} \tag{3.2}$$

In threshold binarization, the most important thing is to choose the appropriate threshold, which is also the difficulty of binarization. Commonly used binarization threshold selection methods include bimodal method, p-parameter method, large law method (Otsu method), maximum entropy threshold method, iterative method and so on. This article uses the Great Law (Otsu method) to select the threshold.

The graythresh (image) function input is an image and the output is a threshold. In this function, a suitable threshold for the picture is found using the maximum interclass variance method. The found threshold satisfies the low (ie, below the selected threshold) portion and the high (ie, above the threshold portion) class variance is greatest.

Then use the imbinarize (image, threshold) Function, input the found threshold, we can convert gray image to binary image. Fig. 3.7 is the binary images after thresholding.

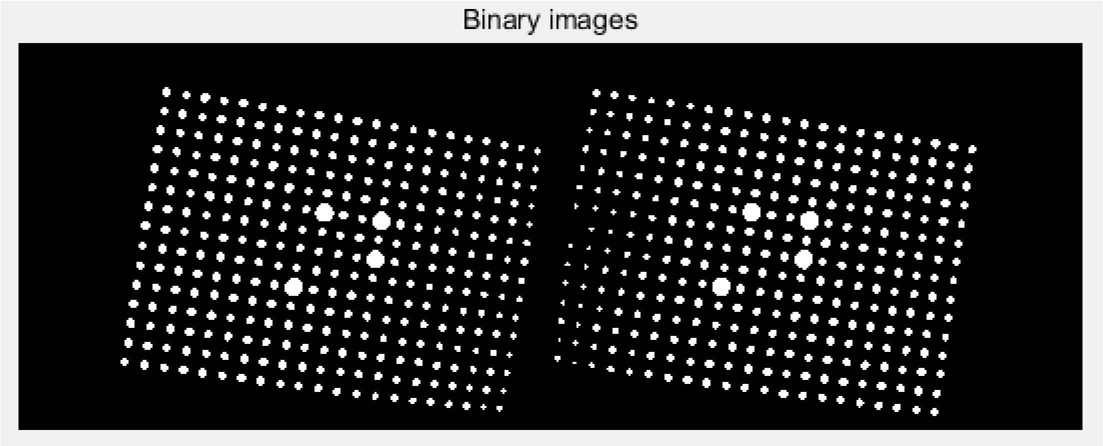


Fig.3.7 Binary image

3.4. Target Recognition and Location

After image binarization, it usually contains multiple regions that need to be extracted separately by markers. A simple and effective way to mark areas in a segmented image is to check for alternating adjacent alternating connections.

In the binary image, the pixel value of the background area is 0, and the pixel value of the target area is 1. The connection area mark allows each individual connection area to form one identification block by marking white pixels (targets) in the mark area. Thereby further obtaining the contour and centroid of the target. In this paper, the target region is extracted by an eight-connected region algorithm.

The ranging system only needs to calculate the distance of the target, so only one pair of matching points is needed. Considering the simple shape of the target, the paper uses the centroid of the target as the matching point. The centroid extraction is performed on the images obtained by the two cameras to complete the matching of the corresponding points.

Figure 3.8 is an extraction of the centroid of the left camera target image and the pixel coordinates of the centroid. Figure 3.9 is an extraction of the centroid of the right camera target image and the pixel coordinates of the centroid.

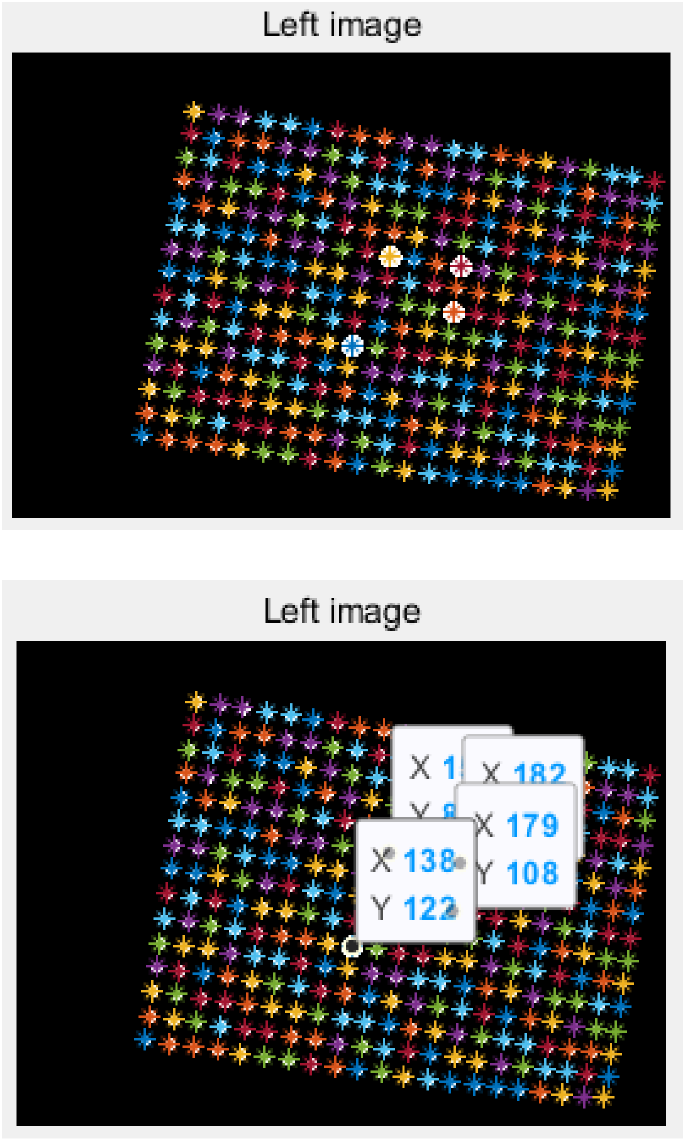
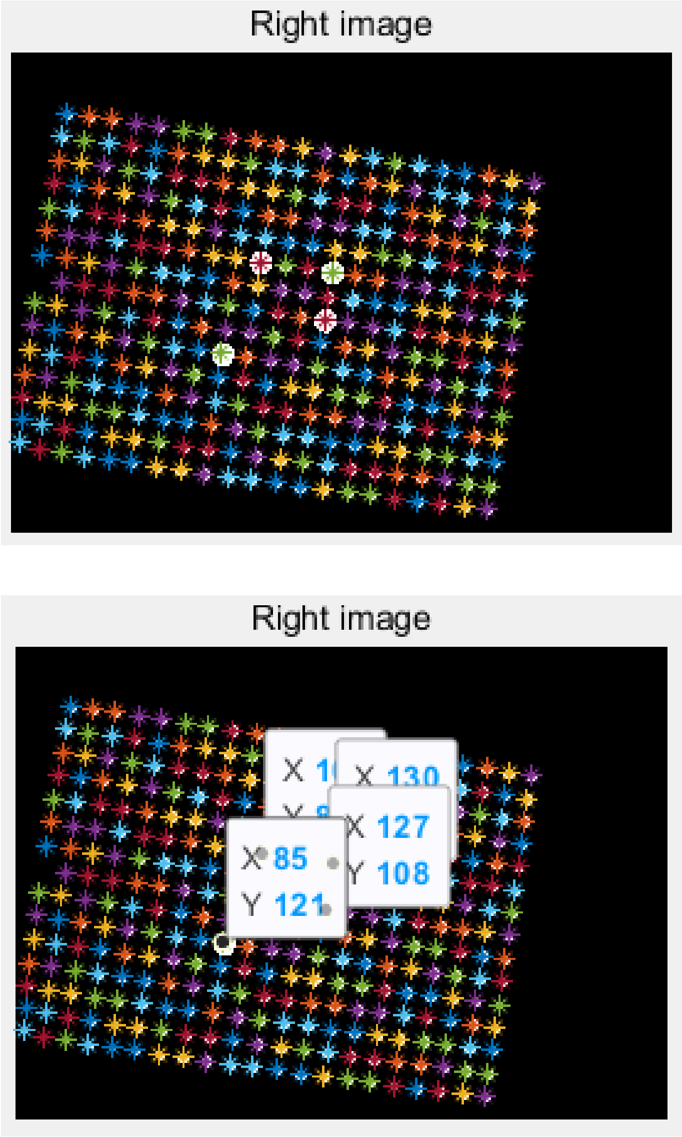


Fig.3.8 Left Camera Image Target Detection



Right image

Right image

Fig.3.9 Right Camera Image Target Detection

3.5. Three-dimensional reconstruction of spatial point coordinates

3.5.1 Projection matrix method

When the binocular cameras are not parallel, the spatial point reconstruction method based on least squares method is used to realize Three-dimensional positioning [16]. The basic model of spatial point 3D reconstruction is shown in Figure 3.10. For a point P in space, the imaging points in the camera Cl and the camera Cr are Pl and Pr, respectively, and the constraints of the two rays of OlPl and OrPr are established between the optical point Ol and Or of the two cameras. Solving the intersection point P is the problem of solving the intersection of spatial lines.

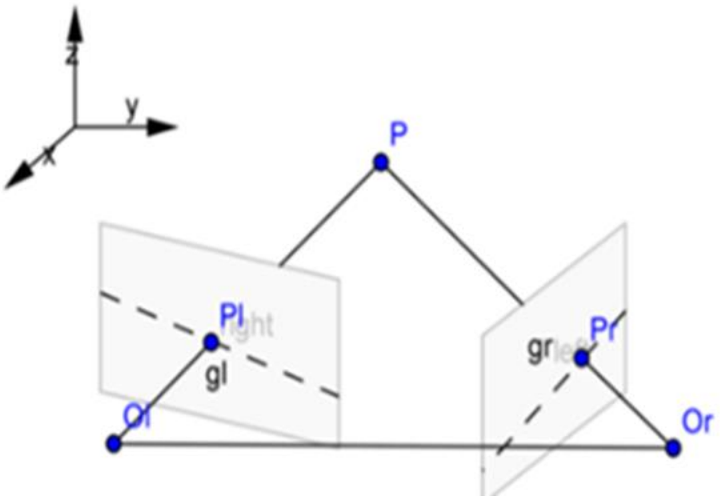


Fig.3.10 Binocular vision 3D information schematic

The calibration of the binocular camera is introduced in the second chapter, and the parameter matrices M1 and M2 of the left and right cameras are obtained. In this chapter, the corresponding feature points Pl and Pr of the two images are obtained. The algorithm for solving P point three-dimensional coordinates is:

$$\mathrm{Z}_C * \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = \begin{bmatrix} {m_{11}}^{l} & {m_{12}}^{l} & {m_{13}}^{l} & {m_{14}}^{l} \\ {m_{21}}^{l} & {m_{22}}^{l} & {m_{23}}^{l} & {m_{24}}^{l} \\ {m_{31}}^{l} & {m_{32}}^{l} & {m_{33}}^{l} & {m_{34}}^{l} \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{3.3}$$

$$\mathrm{Z}_C * \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = \begin{bmatrix} {m_{11}}^{r} & {m_{12}}^{r} & {m_{13}}^{r} & {m_{14}}^{r} \\ {m_{21}}^{r} & {m_{22}}^{r} & {m_{23}}^{r} & {m_{24}}^{r} \\ {m_{31}}^{r} & {m_{32}}^{r} & {m_{33}}^{r} & {m_{34}}^{r} \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{3.4}$$

(u_l, v_l) is the homogeneous coordinate value of the image point P1 on the left image, (u_r, v_r) is the homogeneous coordinate value of the image point Pr on the right image, (X_W, Y_W, Z_W) is the homogeneous coordinate value of the spatial point P, M_l is the projection matrix of the left camera, M_r is the projection matrix of the right camera, Z_C is the depth value of the pixel point in the corresponding depth image [17].

Eliminate Z_c in the two matrices to obtain four linear equations for point P. Expressed in the form of a matrix:

$$\begin{bmatrix} u_l * {m_{31}}^l - {m_{11}}^l & u_l * {m_{32}}^l - {m_{12}}^l & u_l * {m_{33}}^l - {m_{13}}^l \\ v_l * {m_{31}}^l - {m_{21}}^l & v_l * {m_{32}}^l - {m_{22}}^l & u_l * {m_{33}}^l - {m_{23}}^l \\ u_r * {m_{31}}^r - {m_{11}}^r & u_r * {m_{32}}^r - {m_{12}}^r & u_r * {m_{33}}^r - {m_{13}}^r \\ v_r * {m_{31}}^r - {m_{21}}^r & v_r * {m_{32}}^r - {m_{22}}^r & v_r * {m_{33}}^r - {m_{23}}^r \end{bmatrix} * \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} = \begin{bmatrix} {m_{14}}^l - u_l * {m_{34}}^l \\ {m_{24}}^l - v_l * {m_{34}}^l \\ {m_{14}}^r - u_r * {m_{34}}^r \\ {m_{24}}^r - v_r * {m_{34}}^r \end{bmatrix} \tag{3.5}$$

There are three unknowns in the four equations, and theoretically the unique value of the unknown can be calculated. However, linear correlation is almost impossible to occur due to the influence of the errors of the matching points P1 and Pr and the noise in the image. In practical applications, we use the least squares method to solve the three-dimensional coordinates of P. Rewriting as A*X=B, a last square fit solution can be calculated by:

$$\mathbf{X} = (\boldsymbol{A}^{\boldsymbol{T}} * \boldsymbol{A})^{-\mathbf{1}} * \boldsymbol{A}^{\boldsymbol{T}} * \boldsymbol{B} \tag{3.6}$$

3.5.2 Parallax method

When the parameters in the two cameras are the same and placed horizontally, they are called parallel stereo vision models or standard visual models. In this paper, two cameras of the same model are used, and the stereo corrected image also conforms to the parallel condition. Therefore, the method of obtaining the 3D geometric information of the target from two images based on the principle of parallax is adopted， which is simpler and more efficient [18]. As shown in the figure 3.11.

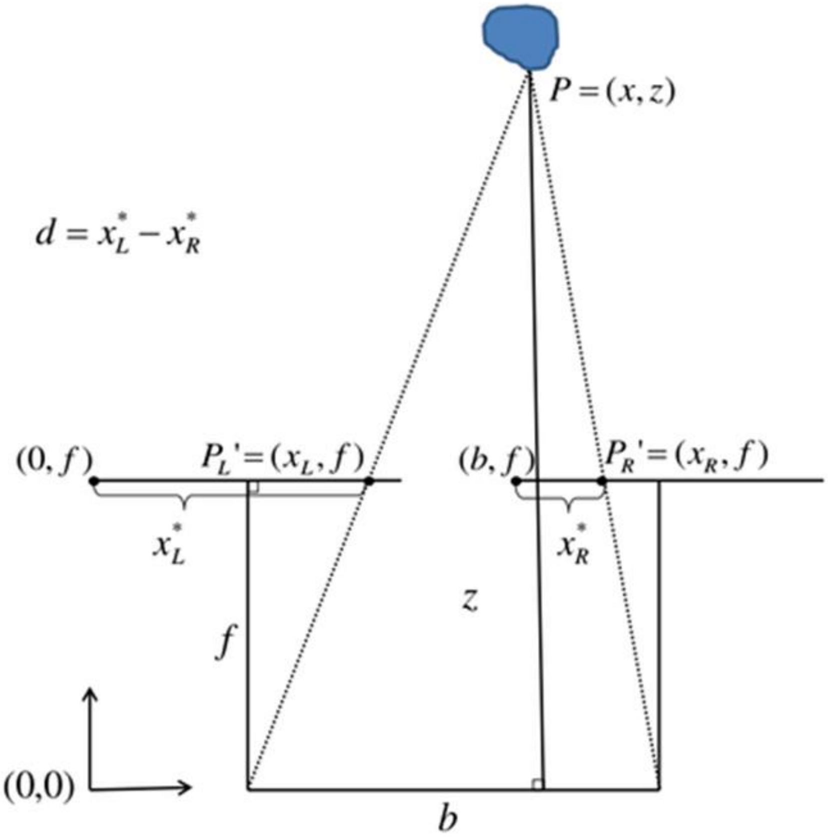


Fig.3.11 Sketch map of parallel binocular stereo vision principle

Binocular stereo vision is based on parallax, and the three-dimensional information is acquired by the trigonometric principle.that is, a triangle is formed between the image plane and the target point of the two cameras.The baseline distance b is distance between the projection centers of the two cameras. The projection points of the spatial point P on the left and right image planes are PL and PR. Since the images of the two cameras are on the same plane, the Y coordinates of the two projection points must be the same, ie Yl = Yr = Y. The following relationship can be obtained from the triangular geometric relationship:

$$\frac{X_l}{X_C} = \frac{f}{Z_C} \tag{3.7}$$

$$\frac{X_r}{X_C} = \frac{f}{Z_C} \tag{3.8}$$

$$\frac{Y}{Y_C} = \frac{f}{Z_C} \tag{3.9}$$

Disparity d=Xl-Xr is known, and the three-dimensional coordinates of the feature point P in the camera coordinate system can be calculated:

$$X_C = \frac{X_l * b}{d} \tag{3.10}$$

$$Y_C = \frac{Y * b}{d} \tag{3.11}$$

$$Z_C = \frac{f * b}{d} \tag{3.12}$$

Any point on the image plane of the left camera can determine the three-dimensional coordinates of the point as long as it can find the corresponding matching point on the image side of the right camera [19]. The three-dimensional coordinates of the target point obtained by parallax methodare shown in Table 2.

Table 2 Target point camera coordinates

	three-dimensional coordinates(mm)
Point 1	[6.7829 45.5581 -342.6538]
Point 2	[-35.6734 39.7020 -342.6538]
Point 3	[-31.2814 11.8859 -342.6538]
Point 4	[28.2007 -8.4478 -336.1886]

The relative distances of the four target points can be calculated from the three-dimensional coordinates of the four target points.

Use a ruler to get the actual length of the target.

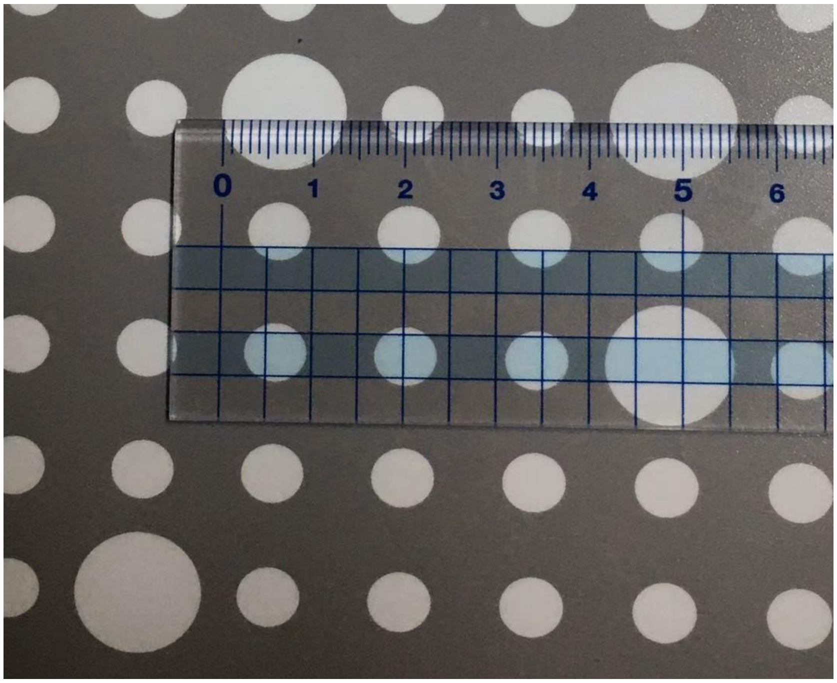


Fig.3.12 L12 Measured length

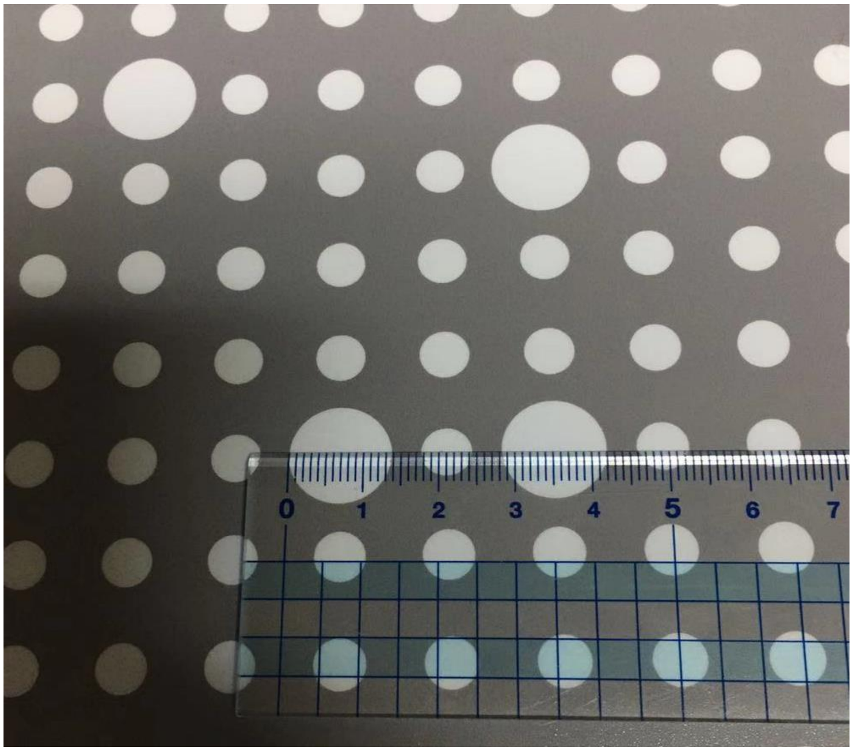


Fig.3.13 L23 Measured length

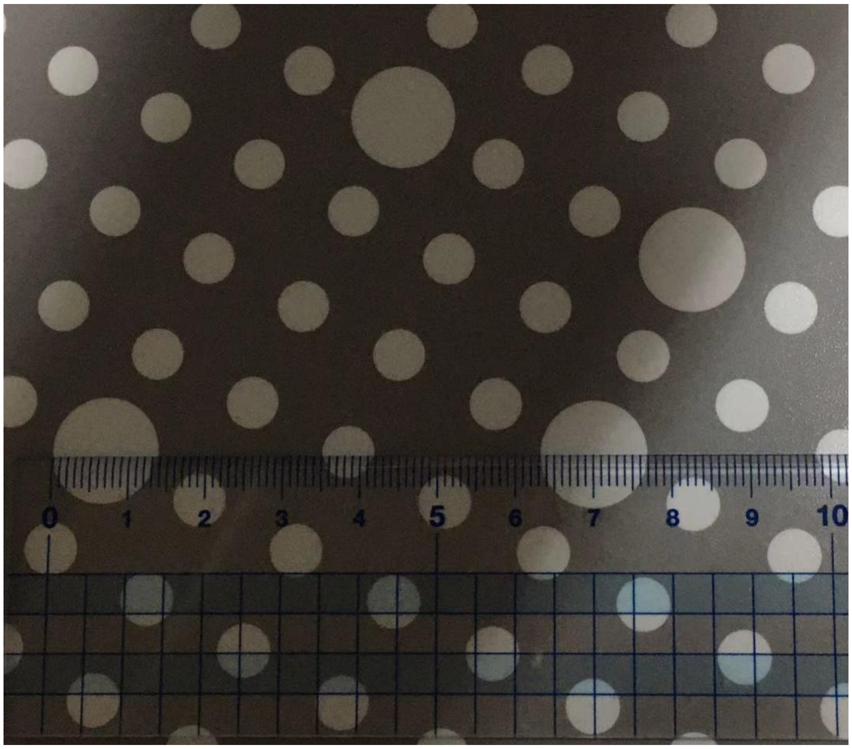


Fig.3.14 L34 Measured length

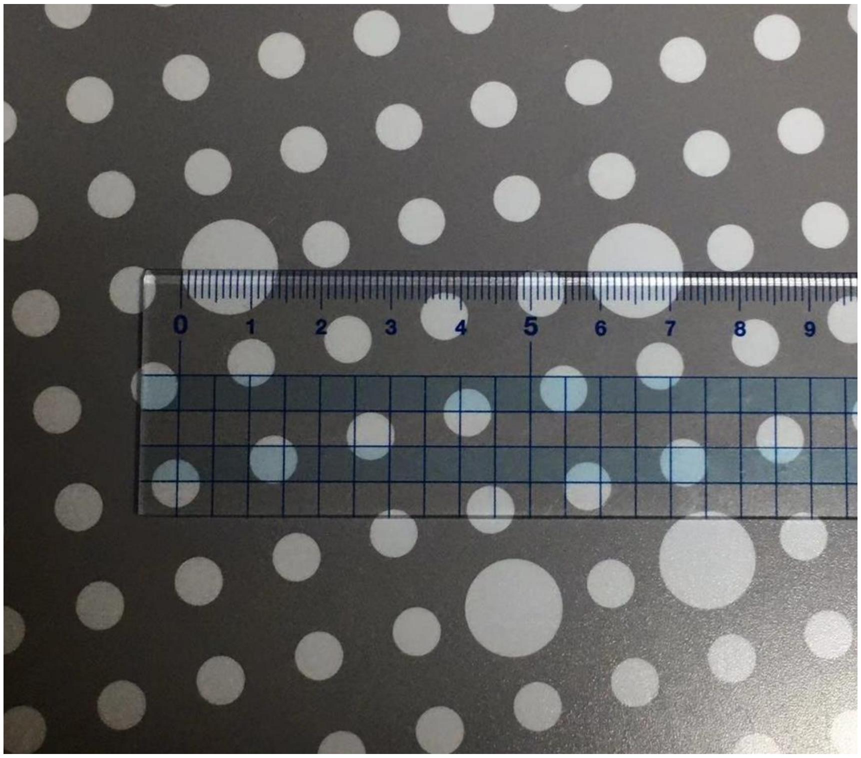


Fig.3.15 L41 Measured length

Table 3 compares the three-dimensional positioning distance with the measured distance.

Table 3 3D ranging and actual distance comparison

Distance	3D ranging length(mm)	Measured length(mm)	Error(mm)
L12	42.86	42.50	0.36
L23	28.16	28.00	0.16
L34	62.86	63.00	0.14
L41	58.10	58.00	0.10

3.6. Conclusion

In this chapter, the geometric feature centroid point of the target is used as the matching point, and based on the parallax method is used to calculate the three-dimensional coordinates, thereby achieving the target ranging. It can be seen from the analysis that the three-dimensional ranging and the actual ranging have certain errors but can meet the actual needs. The main source of error is the matching accuracy of the two images, so future research will optimize the matching algorithm, improve the matching accuracy, and further improve the accuracy of 3D positioning.

CHAPTER IV

HAND-EYE CALIBRATION

4.1. Hand-eye system model

The camera and the end of the robot's hand constitute the hand-eye system. The hand-eye system is divided in to Eye-in-Hand system and Eye-to-Hand system according to the different positions between the camera and the robot.

The camera of Eye-in-Hand system is installed at the end of the robot's hand (End-Efcctor), and the robot moves together randomly in the process of working. The camera of Eye-to-Hand system is installed in a fixed position outside the body of the robot, and the robot does not move together randomly during the working process of the robot.

Eye-in-Hand system is widely used in industrial robots. As the manipulator approaches the target, the distance between the camera and the target decreases, and the absolute error of the camera measurement decreases accordingly. In Eye-in-Hand system, image-based visual control, position-based visual control and hybrid visual control can be used. For image-based visual control, the camera calibration error can be effectively overcome due to the formation of closed-loop in image space, so the accuracy of camera calibration is not required. For position-based visual control, although the camera calibration error can not be effectively overcome in the control system, with the approaching of the target, the absolute error of the measured target position decreases. Even if there are some errors in the camera calibration, it can generally meet the application requirements. Similarly, hybrid vision control has some errors in camera calibration, and can generally meet the application requirements. However, in practical application, the Eye-in-Hand system's vision is changing, which can not guarantee that the target is always in the field of view, sometimes there will be the phenomenon of losing the target.

Eye-to-Hand system has wide application prospects in mobile robots. In this kind of system, when the robot reaches a certain distance from the target and the target is already in the scope of mechanical operation, the robot stops moving towards the target. Thereafter, according to the results of visual measurement, the manipulator moves to the target and operates the target. Generally, when the manipulator moves towards the target and operates, it will occlude the target. Therefore, image-based visual control and hybrid visual control are not suitable for such tasks. Because the camera does not move with the manipulator in the course of moving towards the target, the measurement result of the camera to the target will not change. Because of the distance between the camera and the target, when the camera calibration accuracy is not high, there will be a relatively large absolute error. When the error is large, the manipulator can not reach the target.

In hand-eye vision system, the camera is usually installed on the end-effector of the robot to obtain the position information of the object in space. Then, the decision-making is made by the control system and the robot is ordered to perform the corresponding actions. The key link is to obtain the coordinate conversion relationship between the camera and the end-effector of the robot, that is, the hand-eye calibration. This paper chooses the mode of Eye-in-Hand is used for hand-eye calibration. And Zhang Zhengyou's chessboard calibration method is used for camera calibration. Because hand-eye calibration requires camera calibration results, it is necessary to ensure that the calibration board is fixed and the hand-eye group takes pictures when taking chessboard photos in different directions.

4.2. Hand-eye calibration equation

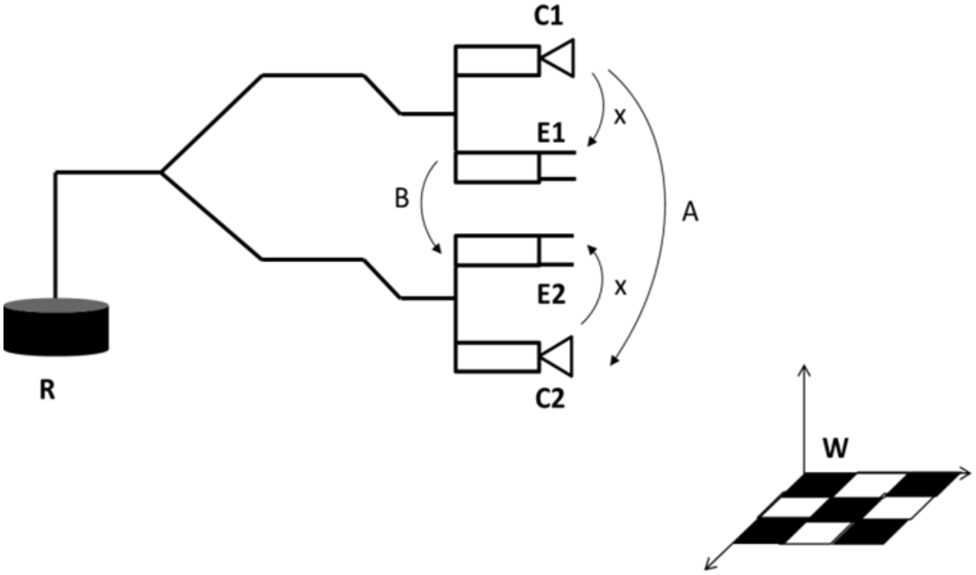


Fig.4.1 Set Up of Deduction Model

W coordinate system is a checkerboard coordinate system for calibration. R coordinate system is a robot base coordinate system. W coordinate system and R coordinate system are fixed and invariable.

Step1: The coordinate transformation from E1 point to E2 point at the end of the manipulator:

At point E1:

$$P_R = H_{E1} * P_{E1} \tag{4.1}$$

$$P_{C1} = H_{w1} * P_W \tag{4.2}$$

At point E2:

$$P_R = H_{E2} * P_{E2} \tag{4.3}$$

$$P_{C2} = H_{w2} * P_W \tag{4.4}$$

Here, H_{w1} and H_{w2} are relative poses of the camera coordinate system and world coordinates. H_{e1} and H_{e2} are relative poses of the end of the robot arm relative to the coordinates of the robot arm. P_w is the space coordinate of the target point in the world coordinate system, and P_R is the space coordinate of the target point in the base coordinate system of the robot arm.

Step2: The camera moves from C1 to C2. Hw1 and Hw2 are the external parameters obtained by camera calibration. That is, the relative pose matrix of the chessboard coordinate system and the camera coordinate system.

According to：

$$P_W = {H_{W2}}^{-1} * P_{C2} \tag{4.5}$$

So:

$$P_{C1} = H_{W1} * {H_{W2}}^{-1} * P_{C2} \tag{4.6}$$

Where:

$$\mathrm{A} = H_{W1} * {H_{W2}}^{-1} \tag{4.7}$$

Step3: The manipulator moves from E1 to E2. He1 and He2 are homogeneous transformation matrices of forward kinematics.

According to：

$$H_{E1} * P_{E1} = H_{E2} * P_{E2} \tag{4.8}$$

So:

$$P_{E1} = {H_{E1}}^{-1} * H_{E2} * P_{E2} \tag{4.9}$$

Where:

$$\mathrm{B} = {H_{E1}}^{-1} * H_{E2} \tag{4.10}$$

Step4: The transformation relationship between the camera and the end of the manipulator is as follows:

According to：

$$P_{C1} = X * P_{E1} \tag{4.11}$$

$$P_{C2} = X * P_{E2} \tag{4.12}$$

Thus:

$$\mathrm{A} * X * P_{E2} = X * B * P_{E2} \tag{4.13}$$

So:

$$\mathrm{A} * \mathrm{X} = \mathrm{X} * \mathrm{B} \tag{4.14}$$

Step5: The matrix relation is:

$$\begin{bmatrix} R_a & T_a \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} R_x & T_x \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_x & T_x \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} R_b & T_b \\ 0 & 1 \end{bmatrix} \tag{4.15}$$

4.3. Hand-eye Matrix method

4.3.1 Rodrigue vector rotation formula

The approach is based on the geometric interpretations of the eigenvalues and eigenvectors of a rotational matrix. Homogeneous transforms [20] can be viewed as the relative position and orientation of a coordinate frame with respect to another coordinate frame. The homogeneous transform is usually denoted as follows:

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4.16)$$

We also denote $[n_x \quad n_y \quad n_z]^T$ as n, $[o_x \quad o_y \quad o_z]^T$ as o, $[a_x \quad a_y \quad a_z]^T$ as a, and $[p_x \quad p_y \quad p_z]^T$ as p. n, o, and a can be interpretated as unit vectors which indicate the x, y, and z directions of coordinate frame T; p can be viewed as the origin of T.

We will refer to the upper-left 3 x 3 submatrix of T as the rotational submatrix since it contains information about the orientation of the coordinate frame. A rotational submatrix can be expressed as a rotation around an arbitrary axis. The matrix representing a right-hand-rule rotation of θ around an axis $[K_x \quad K_y \quad K_z]^T$ is as follows [20]:

$$\mathrm{R}(\mathrm{k}, \theta) = \begin{bmatrix} k_x * k_x * vers(\theta) + \cos(\theta) & k_x * k_y * vers(\theta) - k_z * \sin(\theta) & k_x * k_z * vers(\theta) + k_y * \sin(\theta) \\ k_x * k_y * vers(\theta) + k_z * \sin(\theta) & k_y * k_y * vers(\theta) + \cos(\theta) & k_y * k_z * vers(\theta) - k_x * \sin(\theta) \\ k_x * k_z * vers(\theta) - k_y * \sin(\theta) & k_y * k_z * vers(\theta) + k_x * \sin(\theta) & k_z * k_z * vers(\theta) + \cos(\theta) \end{bmatrix}$$

(4.17)

This matrix is Rodrigues' vector rotation matrix. Where $\mathrm{vers}(\theta) = 1 - \cos(\theta)$. Rodrigues' rotation formula is a new formula for calculating vectors in three-dimensional space after a vector rotates around a given angle of rotation axis. This formula uses the original vector, the axis of rotation and their cross product as the frame to represent the vector after rotation.

We get the following equation easily from equation (4.17):

$$\cos(\theta) = \frac{1}{2}(n_x + o_y + a_z - 1) \qquad (4.18)$$

We can now calculate k using θ .The set of equations used depends on which of n_x, o_y, or a_z is maximum.

If $n_x > o_y$ and $n_x > a_z$:

$$k_x = sgn\left(o_z - a_y\right)\sqrt{\frac{n_x - \cos(\theta)}{vers(\theta)}} \tag{4.19}$$

$$k_y = \frac{n_y + o_x}{2 * k_x * vers(\theta)} \tag{4.20}$$

$$k_z = \frac{a_x + n_z}{2 * k_x * vers(\theta)} \tag{4.21}$$

If $o_y > n_x$ and $o_y > a_z$:

:

$$k_y = sgn(a_x - n_z)\sqrt{\frac{o_y - \cos(\theta)}{vers(\theta)}} \tag{4.22}$$

$$k_x = \frac{n_y + o_x}{2 * k_y * vers(\theta)} \tag{4.23}$$

$$k_z = \frac{o_z + a_y}{2 * k_y * vers(\theta)} \tag{4.24}$$

If $a_z > n_x$ and $a_z > o_y$:

$$k_z = sgn(n_y - o_x)\sqrt{\frac{a_z - \cos(\theta)}{vers(\theta)}} \tag{4.25}$$

$$k_x = \frac{a_x + n_z}{2 * k_z * vers(\theta)} \tag{4.26}$$

$$k_y = \frac{o_z + a_y}{2 * k_z * vers(\theta)} \tag{4.27}$$

k_a and k_b can be obtained by matrix A and matrix B.

According to:

$$k_X = k_b * k_a \tag{4.28}$$

$$\theta_X = \mathrm{atan}(|k_b * k_a|, k_b \cdot k_a) \tag{4.29}$$

We can get:

$$R_X = R(k_X\ , \theta_X) \tag{4.30}$$

According to:

$$R_a * T_X + T_a = R_X * T_b + T_X \tag{4.31}$$

We can get:

$$(R_a - I) * T_X = R_X * T_b - T_a \tag{4.32}$$

Rewriting as:

$$\mathrm{E} * T_X = F \tag{4.33}$$

Least squares fit solution can be calculated by:

$$T_X = (E^T * E)^{-1} * E^T * F \tag{4.34}$$

4.3.2 Matrix Linear Algorithms

Shiu [21] proposed a mathematical solution to the problem of hand-eye (sensor) calibration, and summed it up as AX = XB, where $\mathrm{A} = \mathrm{A}_1 * \mathrm{A}_2^{-1}$ is the relative relationship between the camera coordinate system before and after moving, and $\mathrm{B} = \mathrm{B}_1^{-1} * \mathrm{B}_2$ is the transformation relationship between the robot end-effector coordinate system before and after moving, as shown in Fig.4.2. A_i is the conversion from camera to world coordinate system, which is determined by the external parameters of camera calibration; B_i is the conversion from base coordinate system of manipulator to end-effector of manipulator, which is calculated by forward kinematics.

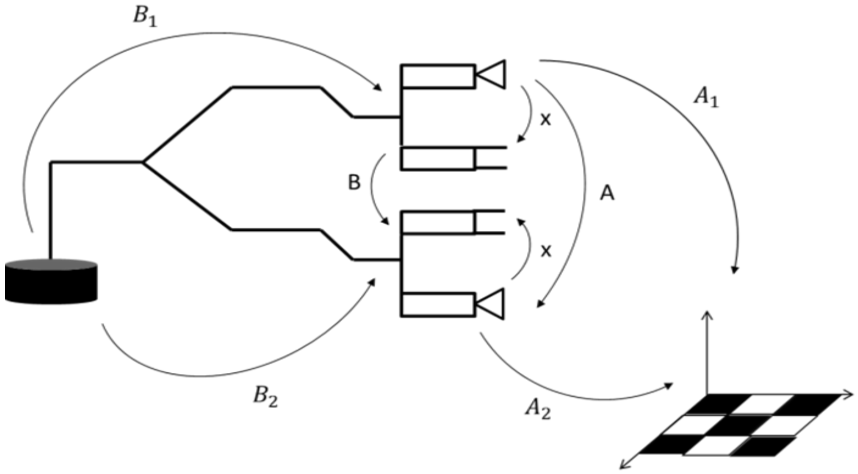


Fig.4.2 Set Up of Modeless Calibration

All homogeneous transformation matrices are expressed in the following forms:

$$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \tag{4.35}$$

So we can get:

$$\begin{cases} R_a * R_x = R_x * R_b \\ R_a * t_x + t_a = R_x * t_b + t_x \end{cases} \tag{4.36}$$

The linear operator VEC and tensor product are used to linearize the matrix. The VEC operator arranges the matrix elements of $\mathrm{m} \times \mathrm{n}$ into a vector whose length is $\mathrm{m} \cdot \mathrm{n}$ according to row priority.

$$\mathrm{vec(M)} = (M_{11}, M_{12} \dots M_{1n}, M_{21}, \dots M_{mn})^T \tag{4.37}$$

The tensor product satisfies the equation $\mathrm{vec(ABC)} = (\mathrm{A} \otimes \mathrm{C}^{\mathrm{T}})\mathrm{vec(B)}$. The product of matrices is transformed into the product of matrices and vectors by direct product.

According to:

$$\begin{cases} R_a * R_x * I_3 - I_3 * R_x * R_b = 0 \\ I_3 * R_x * t_b + (I_3 - R_a)t_x = t_a \end{cases} \tag{4.38}$$

We can get:

$$\begin{cases} (R_a \otimes I_3 - I_3 \otimes {R_b}^T)vec(R_x) = 0_9 \\ (I_3 \otimes {t_b}^T)vec(R_x) + (I_3 - R_a)t_x = t_a \end{cases} \tag{4.38}$$

Where:

$$\begin{bmatrix} (R_{a1} \otimes I_3 - I_3 \otimes {R_{b1}}^T) \\ (R_{a2} \otimes I_3 - I_3 \otimes {R_{b2}}^T) \end{bmatrix} vec(R_x) = 0_9 \tag{4.39}$$

Since the rotation matrix R_x is not 0, the vector $vec(R_x)$ is not 0, so there is at least one eigenvalue (singular value) of 0 in the left coefficient matrix.

The coefficient matrix obtains the eigenvector v corresponding to the singular value 0 by singular value decomposition. The scale factor of the solution of the equation and v is a, and the scale factor is substituted into the equation to be solved.

We can get:

$$\begin{cases} (I_3 \otimes {t_{b1}}^T) * v * a + (I_3 - R_{a1})t_x = t_{a1} \\ (I_3 \otimes {t_{b2}}^T) * v * a + (I_3 - R_{a2})t_x = t_{a2} \end{cases} \tag{4.40}$$

Where:

$$\begin{cases} M_1 = (I_3 \otimes {t_{b1}}^T) * v \\ M_2 = (I_3 \otimes {t_{b2}}^T) * v \end{cases} \qquad \begin{cases} N_1 = I_3 - R_{a1} \\ N_2 = I_3 - R_{a2} \end{cases} \tag{4.41}$$

The matrix is transformed into a linear system of equations AX=B for calculation.

$$\mathrm{A} = \begin{bmatrix} M_1 & N_1 \\ M_2 & N_2 \end{bmatrix} \qquad \mathrm{X} = [a \quad t_{x1} \quad t_{x2} \quad t_{x3}\ ]^T \qquad B = [t_{a1}^T \quad t_{a2}^T]^T \tag{4.42}$$

Least squares fit solution can be calculated by:

$$\mathrm{X} = (A^T * A)^{-1} * A^T * B \tag{4.43}$$

The first element of the vector X is a constant scale factor a, and the next three elements are the components of the hand-eye translation matrix.

According to:

$$\mathrm{vec}(R_x) = v * a \tag{4.44}$$

We can get the hand-eye rotation matrix:

$$R_x = (vec(v * a))^T \tag{4.45}$$

4.4. Computational experiment and analysis

Tsai [22] pointed out that in order to uniquely determine the components of the hand-eye matrix, at least two sets of motion with non-parallel rotation axes are needed. The camera is calibrated to obtain 10 sets of external parameters. Since the first 3 pictures have too much error, 6 position transformations are calculated from the last 7 sets of external parameters. One hand-eye matrix can be calculated for every 2 position transformations, and 15 hand-eye matrices can be calculated by combining the 6 position transformations in pairs. The average of the 15 hand-eye matrices is calculated as the solution of the hand-eye matrix. The two methods are respectively solved and the obtained result is substituted into $\mathrm{error} = \| \mathrm{A_i x - xB_i} \|$ to obtain the residual value of the hand-eye equation. The error values are shown in Table 4.

Table 4 Error of hand-eye matrix calculation

Error	Formula	Matrix Linear Algorithms	Rodrigues vector rotation formula
err1	$\| \mathrm{A_1x - xB_1} \|$	0.0016	0.0015
err2	$\| \mathrm{A_2x - xB_2} \|$	0.0012	0.0016

The linear form of the hand-eye matrix of the robot is deduced by the direct product of the matrix, and the linear solution is obtained by the least square method [23]. The rotational part of the solution is orthogonalized by the RO-drigues formula. The calulation experiment proves that the error introduced by the orthogonalization is permissible in most cases.

In the previous chapter, we calculated the spatial coordinates of the target point in the camera coordinate system through the principle of vision. Therefore, the spatial coordinates of the target point in the coordinate system of the end of the robot arm can be calculated through the hand-eye matrix.

$$\begin{bmatrix} X_T \\ Y_T \\ Z_T \\ 1 \end{bmatrix} = \mathrm{x} * \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} \tag{4.46}$$

$[X_C \quad Y_C \quad Z_C]^T$ is the spatial coordinates of the target point in the camera coordinate system. $[X_T \quad Y_T \quad Z_T]^T$ is the spatial coordinates of the target point in the coordinate system of the end of the robot arm.

4.5. Conclusion

In this chapter, the hand-eye calibration methods are studied in detail, and two aspects are summarized. Firstly, the general steps of Rodrigue vector rotation formula robot hand-eye calibration method are summarized, which makes the calibration process easier to understand. Secondly, an improved hand-eye calibration algorithm is presented by solving some problems encountered in the Rodrigue vector rotation formula method, and the principle and steps of the algorithm are given in detail. This improved algorithm deduces the linear equation of the hand-eye conversion matrix of the robot by using the direct product of the matrix and the eigenvector of the matrix. It does not need other mathematical tools and simplifies the calculation. Both methods have some error and the error is within the allowable range.

CHAPTER V

DISCUSSIONS

In this thesis, I focus on three-dimensional target localization, especially the coordinates of the target in the robot arm coordinate system by hand-eye calibration.

In Chapter II, the camera imaging model is established, the camera distortion is analyzed and the internal and external parameters of the camera are calculated by MATLAB stereo correction.

In Chapter III, the pixel coordinates of the corresponding target points of the left and right images are obtained by stereo matching of the camera, and the three-dimensional coordinates of the target are calculated based on the least squares method.

In Chapter IV, the hand-eye matrix is calculated by two methods, and the coordinates of the target in the tool coordinate system are obtained, which provides data support for the next path planning and target capture.

REFERENCE

[1]. Garcia E, Jimnez M.A, De Santos P.G, Armada M. The evolution of robotics research [J].IEEE Robots & Automation Magazine, 2007, 14(1), 90-103

[2]. Hill J.Real Time Control of a Robot with a Mobile Camera [J]. Robotics and Automation, 1996, 12(5) pp.651-670

[3]. Levine S, Pastor P, Kriz hevsky A, et al. Learning Hand -Eye Coordination for Robotic Grasping with Large-Scale Data Collection[J]. International Journal of Rob ot ics Research, 2016(10):173-184.

[4]. J Wang, et al. Reasearch on object recognition of intelligent robot base on binocular vision [J]. Applied Mechanics and Materials, 2012, 127:300-304

[5]. Kennedy J, Eberhart R C, Particle Swarm Optimization [C]. Proceedings of the IEEE International Conference on Neural Networks.2005:1945-1948

[6]. Ryberg A, Lennartson B, Christiansson A K, et al, Analysis and evaluation of a general camera model [J]. Computer Vision & Image Understanding, 2011, 115(11): 1503-1515

[7]. R.Hartley and A, Zisserman, Multiple View Geometry in Computer Vision [J]. Cambridge University Press, 2011

[8]. Ricolfe-Viala C, Sánchez-Salm er ón A J. Using the camera pin-hole model restrictions to calibrate the lens distortion model [J]. Optics & Laser Technology, 2011, 43(6):996-1005

[9]. P.Corke, P.Dunn, Real-time stereopsis using FPGAs, Proceedings IEEE. TENCON- Speech and Image Technologies for Computing and Telecommunications, 2007:235-238

[10]. Z. Zhang, A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 2000: pp 1330-1334

[11]. K.Konolige, Small vision systems, Hardware and implementation, Proceedings Eighth International Symposium Robotics Research, 2007

[12]. O.Veksler, Fast variable window for stereo correspondence using integral images, Computational Vision and Active Perception Laboratory(CVAP), 2009

[13]. Levine S, Pastor P, Krizhevsky A, et al. Learning Hand-Eye Coordination for Robotic Grasping with Large-Scale Data Collection [J]. International Journal of Robotics Research, 2016(10) pp: 173-184

[14]. J Aguilara, F. Torresa and MA Lopea, "Stereoscopic Vision for 3D Measurement: Accuracy Analysis, Calibration and Industrial Applications", Measurements, Vol. 18, No. 4, pp. 193-200, 1996.

[15]. G Egnal, R P Wildes. Detecting binocular half-occlus ions empirical comparisons of five approaches [J], IEEE Transactio ns on Pattern Analysis and Machine Intelligence, 2002, 24(8): 1127-1133

[16]. Zhengyou Zhang, Gang Xu. A General Expression of the Fundamental Matrix for Both Perspective and Affine Cameras [J], Proc of 15th Internatio nal Joint Conference on Artificial Intelligence, 1997: 1502-1507

[17]. Henry P, Krainin M, Herbst E, et al. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indo or enviro nments [J]. Internatio nal Journal of Rob otics Research, 2013, 31(5): 647-663.

[18]. Gay D, Levis P Culler D. Software design patterns for TinyOS [J]. ACM Transcations on Embedded Computing Systems (TECS)-Special Section LCTES S, 2007

[19]. J Aguilara, F. Torresa and MA Lopea, "Stereoscopic Vision for 3D Measurement: Accuracy Analysis, Calibration and Industrial Applications", Measurements, Vol. 18, No. 4, pp. 193-200, 1996.

[20]. R. P. Paul, Robot Manipulators: Mathematics, Programming and Confrol. Cambridge, MA: MIT Press, 1981.

[21]. Y. C. Shiu and S. Ahmad. CaIibration of wrist - mounted robotic sensors by soIving homogeneous transform eguations of the form AX= XB, IEEE Transaction of Robotics and Automation. VoI.5, No.l, 1989, pp.l6 – 27

[22]. ROGER Y.TSAI, R. K. Lenz, A New Technigue for FuIIy Autonomous and Efficient 3D Robotics Hand/Eye CaIibration, IEEE Transactions on Robotics and Automation, VoI.5. No.3. June, l989, pp.345 - 358

[23]. Levine S, Pastor P, Krizhevsky A, et al. Learning Hand-Eye Coordination for Robotic Grasping with Large-Scale Data Collection [J]. International Journal of Robotics Research, 2016(10) pp: 173-184

Appendix:

1. Hand-eye matrix by Rodrigues vector rotation formula

```
  a=[];
  b=[];
  X1=[];
  Error1=[];
  Error2=[];
  x=zeros(4,4);
  err1=zeros(4,4);
  err2=zeros(4,4);

t=importdata('C:\\Users\\Administrator\\Desktop\\Eye_Hand\\Eye_Hand\\HandEyeC
alibrate\\robot1.txt');

for i=4:9

TI=[stereoParams.CameraParameters1.RotationMatrices(:,:,i);stereoParams.Camer
aParameters1.TranslationVectors(i,:)/1000];

TJ=[stereoParams.CameraParameters1.RotationMatrices(:,:,i+1);stereoParams.Cam
eraParameters1.TranslationVectors(i+1,:)/1000];
     Ti=[TI';0  0  0  1];
     Tj=[TJ';0  0  0  1];
     A=Ti*(inv(Tj));
     a=[a;A];

     Zi=t(i,6)*pi/180;
     Yi=t(i,5)*pi/180;
     Xi=t(i,4)*pi/180;

     ti1=(t(i,1)/1000);
     ti2=(t(i,2)/1000);
     ti3=(t(i,3)/1000);
     ti=[ti1  ti2  ti3]';

     RZi=[cos(Zi)  -sin(Zi)   0;
      sin(Zi)   cos(Zi)    0;
         0        0        1];

     RYi=[cos(Yi)    0     sin(Yi);
         0      1        0;
      -sin(Yi)   0     cos(Yi)];

     RXi=[1          0        0;
       0       cos(Xi)   -sin(Xi);
       0       sin(Xi)    cos(Xi)];

     Ri=RZi*RYi*RXi;
     Zj=t(i+1,6)*pi/180;
     Yj=t(i+1,5)*pi/180;
     Xj=t(i+1,4)*pi/180;

     tj1=(t(i+1,1)/1000);
     tj2=(t(i+1,2)/1000);
     tj3=(t(i+1,3)/1000);
     tj=[tj1 tj2 tj3]';

     RZj=[cos(Zj)  -sin(Zj)   0;
      sin(Zj)   cos(Zj)    0;
         0        0        1];

     RYj=[cos(Yj)    0     sin(Yj);
         0        1        0;
      -sin(Yj)    0     cos(Yj)];

    RXj=[1          0        0;
       0       cos(Xj)   -sin(Xj);
       0       sin(Xj)    cos(Xj)];

    Rj=RZj*RYj*RXj;

    Hi=[Ri ti;0 0 0 1];
    Hj=[Rj tj;0 0 0 1];
    B=(inv(Hi))*(Hj);
    b=[b;B];
end

 for g=1:6
     for k=g+1:6
  A1=a(4*g-3:4*g,1:4);
  A2=a(4*k-3:4*k,1:4);
  B1=b(4*g-3:4*g,1:4);
  B2=b(4*k-3:4*k,1:4);

  Date = [A1 B1 A2 B2];
RovVecAssm = [];
for i = 1:4
A = Date(1:4,(1+(i-1)*4):4*i);
DuiJiaoVector = [A(1,1),A(2,2),A(3,3)];
MaxDuiJiaoVector = max (DuiJiaoVector);
CosSaita = (A(1,1)+A(2,2)+A(3,3)-1)/2;
VersSaita = 1-CosSaita;
if A(1,1) == MaxDuiJiaoVector
    kx = (sign(A(3,2)-A(2,3)))*sqrt((A(1,1)-CosSaita)/VersSaita);
    ky = (A(2,1)+A(1,2))/(2*kx*VersSaita);
    kz = (A(1,3)+A(3,1))/(2*kx*VersSaita);
    RovoteVector = [kx;ky;kz];
elseif A(2,2) == MaxDuiJiaoVector
    ky = sign(A(1,3)-A(3,1))*sqrt((A(2,2)-CosSaita)/VersSaita);
    kx = (A(2,1)+A(1,2))/(2*ky*VersSaita);
    kz = (A(3,2)+A(2,3))/(2*ky*VersSaita);
    RovoteVector = [kx;ky;kz];
else
    kz = sign(A(2,1)-A(1,2))*sqrt((A(3,3)-CosSaita)/VersSaita);
    kx = (A(1,3)+A(3,1))/(2*kz*VersSaita);
    ky = (A(3,2)+A(2,3))/(2*kz*VersSaita);
    RovoteVector = [kx;ky;kz];
end
    RovVecAssm = [RovVecAssm  RovoteVector];
end

for j=1:2:3
XRotVec = cross(RovVecAssm(:,j+1)',RovVecAssm(:,j)');
WAngle = atan2(abs(norm(XRotVec)),RovVecAssm(:,j+1)'*RovVecAssm(:,j));
XRotVec = abs(XRotVec/norm(XRotVec));

VersSaita = 1-cos(WAngle);
Rot11 = (XRotVec(1)^2)*VersSaita+cos(WAngle);
Rot21 = (XRotVec(1)*XRotVec(2))*VersSaita+sin(WAngle)*XRotVec(3);
Rot31 = (XRotVec(1)*XRotVec(3))*VersSaita-sin(WAngle)*XRotVec(3);

Rot12 = (XRotVec(1)*XRotVec(2))*VersSaita-sin(WAngle)*XRotVec(3);
Rot22 = (XRotVec(2)^2)*VersSaita+cos(WAngle);
Rot32 = (XRotVec(2)*XRotVec(3))*VersSaita+sin(WAngle)*XRotVec(1);

Rot13 = (XRotVec(1)*XRotVec(3))*VersSaita+sin(WAngle)*XRotVec(2);
Rot23 = (XRotVec(2)*XRotVec(3))*VersSaita-sin(WAngle)*XRotVec(1);
Rot33 = (XRotVec(3)^2)*VersSaita+cos(WAngle);

Rx=[Rot11 Rot12 Rot13; Rot21 Rot22 Rot23; Rot31 Rot32 Rot33];
Rx = Rx/norm(Rx);
end

RA1 = A1(1:3,1:3);
RA2 = A2(1:3,1:3);
PA1 = A1(1:3,4);
PA2 = A2(1:3,4);
PB1 = B1(1:3,4);
PB2 = B2(1:3,4);
I=eye(3);
E = [RA1-I;RA2-I];
F = [Rx* PB1-PA1;Rx*PB2-PA2];
Px=(inv(E'*E))*E'*F;
X=[Rx Px;0 0 0 1];
X1=[X1;X];
x=x+X;
     end
 end
x=x/15

 for j=1:6
   for k=j+1:6

    A1=a(4*j-3:4*j,1:4);
    A2=a(4*k-3:4*k,1:4);
    B1=b(4*j-3:4*j,1:4);
    B2=b(4*k-3:4*k,1:4);

    error1=A1*x-x*B1;
    error2=A2*x-x*B2;
    Error1=[Error1;error1];
    Error2=[Error2;error2];

    err1=err1+error1;
    err2=err2+error2;

    end
 end


  er1=err1/15
  er2=err2/15
  er1= norm(er1)
  er2= norm(er2)
```

2. Hand-eye matrix by matrix linear algorithms

```
  a=[];
  b=[];
  X1=[];
  Error1=[];
  Error2=[];
  x=zeros(4,4);
  err1=zeros(4,4);
  err2=zeros(4,4);

t=importdata('C:\\Users\\Administrator\\Desktop\\Eye_Hand\\Eye_Hand\\HandEyeC
alibrate\\robot1.txt');
for i=4:9

TI=[stereoParams.CameraParameters1.RotationMatrices(:,:,i);stereoParams.Camer
aParameters1.TranslationVectors(i,:)/1000];

TJ=[stereoParams.CameraParameters1.RotationMatrices(:,:,i+1);stereoParams.Cam
eraParameters1.TranslationVectors(i+1,:)/1000];
     Ti=[TI';0 0 0 1];
     Tj=[TJ';0 0 0 1];
     A=Ti*(inv(Tj));
     a=[a;A];

     Zi=t(i,6)*pi/180;
     Yi=t(i,5)*pi/180;
     Xi=t(i,4)*pi/180;

     ti1=(t(i,1)/1000);
     ti2=(t(i,2)/1000);
     ti3=(t(i,3)/1000);
     ti=[ti1 ti2 ti3]';

  RZi=[cos(Zi)  -sin(Zi)   0;
      sin(Zi)   cos(Zi)    0;
         0        0        1];

  RYi=[cos(Yi)    0     sin(Yi);
         0      1        0;
      -sin(Yi)   0     cos(Yi)];

  RXi=[1          0        0;
       0       cos(Xi)   -sin(Xi);
       0       sin(Xi)    cos(Xi)];

   Ri=RZi*RYi*RXi;



     Zj=t(i+1,6)*pi/180;
     Yj=t(i+1,5)*pi/180;
     Xj=t(i+1,4)*pi/180;

     tj1=(t(i+1,1)/1000);
     tj2=(t(i+1,2)/1000);
     tj3=(t(i+1,3)/1000);
     tj=[tj1 tj2 tj3]';

  RZj=[cos(Zj)  -sin(Zj)   0;
      sin(Zj)   cos(Zj)    0;
         0        0        1];

  RYj=[cos(Yj)    0     sin(Yj);
         0        1        0;
      -sin(Yj)    0     cos(Yj)];

  RXj=[1          0        0;
       0       cos(Xj)   -sin(Xj);
       0       sin(Xj)    cos(Xj)];

   Rj=RZj*RYj*RXj;

   Hi=[Ri ti;0 0 0 1];
   Hj=[Rj tj;0 0 0 1];
   B=(inv(Hi))*(Hj);
   b=[b;B];
end
for j=1:6
     for k=j+1:6

  A1=a(4*j-3:4*j-1,1:3);
  A2=a(4*k-3:4*k-1,1:3);
  B1=b(4*j-3:4*j-1,1:3);
  B2=b(4*k-3:4*k-1,1:3);
```

```
 ta1=a(4*j-3:4*j-1,4);
 ta2=a(4*k-3:4*k-1,4);
 tb1=b(4*j-3:4*j-1,4);
 tb2=b(4*k-3:4*k-1,4);

 C = [kron(A1,eye(3))-kron(eye(3),B1');kron(A2,eye(3))-kron(eye(3),B2')];
 [U,S,V] = svd(C);
 EigVector = V(:,length(V(1,:)));

 M1 = (kron(eye(3),(tb1')))*EigVector;
 M2 = (kron(eye(3),(tb2')))*EigVector;
 N1 = eye(3)-A1;
 N2 = eye(3)-A2;
 CoffMat = [M1 N1;M2 N2];
 ConsNum = [ta1' ta2']';
 Y = (inv(CoffMat'*CoffMat))*CoffMat'*ConsNum;
 EigVector = orth(EigVector);
 vec = Y(1)*EigVector;
 Rx=reshape(vec,3,3)';
 Rx = Rx/norm(Rx);
 Tx= [Y(2);Y(3);Y(4)];
 X=[Rx Tx;0 0 0 1];
 X1=[X1;X];
 x=x+X;
    end
end

  x=x/15

for j=1:6
  for k=j+1:6

   a1=a(4*j-3:4*j,1:4);
   a2=a(4*k-3:4*k,1:4);
   b1=b(4*j-3:4*j,1:4);
   b2=b(4*k-3:4*k,1:4);

   error1=a1*x-x*b1;
   error2=a2*x-x*b2;
   Error1=[Error1;error1];
   Error2=[Error2;error2];
   err1=err1+error1;
   err2=err2+error2;
   end
end
 er1=err1/15
 er2=err2/15
 er1= norm(er1)
 er2= norm(er2)
```