Development of a sea urchin recognition system using underwater stereo vision

Master’s Thesis

DEVELOPMENT OF A SEA URCHIN RECOGNITION SYSTEM USING UNDERWATER STEREO VISION

September 2021

Graduate School of Marine Science and Technology
Tokyo University of Marine Science and Technology
Master’s Course of Marine System Engineering

CHAI JIAYU

Master’s Thesis

DEVELOPMENT OF A SEA URCHIN RECOGNITION SYSTEM USING UNDERWATER STEREO VISION

September 2021

Graduate School of Marine Science and Technology

Tokyo University of Marine Science and Technology

Master’s Course of Marine System Engineering

CHAI JIAYU

Abstract

Since the Great East Japan Earthquake, severe isoyake has occurred in some areas of the Shizugawa Bay in Miyagi Prefecture. To protect the algae and prevent the sea urchin population from continuing to grow, we developed a stereo camera system for large ROV, which mainly uses Deep Learning for pattern matching, accelerometer for camera angle correction, and laser size correction to measure the size and distance of sea urchins, and thus perform automatic capture of sea urchins. The captured sea urchins are not fat and most of them are thrown away. If the captured sea urchins are put into the tank for further breeding, they can grow into fatty that can be eaten. By establishing a system to re-culture and sell sea urchins with little flesh, we can effectively use aquatic resources and contribute to sustainable farming. Artificial culture of sea urchins requires uniform size of sea urchins, otherwise cannibalism and slow growth will occur. In addition, it is difficult to specify the amount of bait to be fed, so the aquaculture tanks are prone to residue decay, leading to deterioration of water quality, and sea urchins are highly susceptible to bald sea urchin and other special sea urchins' diseases. Therefore, we need to develop special system for aquaculture farms. Considering the cost of aquaculture and the requirement to realize the recognition of the same size of sea urchin, this paper develops a simple small underwater stereo camera system, which does not rely on other hardware, and only realizes the function through code.

Sea urchin ranging firstly needs to calibrate the binocular camera, so we can obtain the internal and external parameters of the camera for image rectification in subsequent operations and to realize the coordinate system conversion of image to world. To conveniently obtain the parameters of the binocular camera and avoid the complexity of parameter transfer between cross-platforms, this paper adopts the Zhang's algorithm based on OpenCV-Python with a chessboard grid for calibration. Comparing the parameters obtained from calibration experiments for specific binocular cameras, this paper obtains parameters with better depth error accuracy.

ROV needs to capture sea urchins, so the binocular camera needs to recognize the sea urchins and obtain the image coordinate values of the sea urchins by the sea urchin classifier previously studied in the laboratory to recognize the sea urchins in the pictures. To find the corresponding matching detection frames from multiple detection frames in the pictures captured by the left and right cameras at the same time, this paper applies machine learning (Template matching) or neural network method (Siamese network matching) to calculate the similarity between two detection frames. By comparing the results of the above two methods in terms of error results and time consuming, template matching method can complete the similarity calculation in a shorter time while maintaining a certain accuracy, so the machine learning is chosen in this paper.

The conversion from pixel information to actual distance requires finding matching points in the left and right images, and there are two stereo matching methods, namely machine learning or deep learning. However, compared with deep learning, machine learning can change the parameters according to the effect at any time and does not require a high-performance computer, so the semi-global block matching (SGBM) algorithm is used in this paper. In this paper, we first conducted a ranging experiment on sea urchins'picture in the air environment and obtained better image processing parameters. Further underwater experiments are carried out on the system. Through experimental comparison, the system developed in this paper can achieve better errors.

Contents

1. Introduction

1.1. Isoyake

Before the Great East Japan Earthquake, Shizugawa Bay in Miyagi Prefecture was a high-quality algae farm and thus a breeding ground for other fish species such as abalone and silver salmon, as well as specialties such as octopus and ascidian. However, since the Great East Japan Earthquake, some areas of Shizukawa Bay in Miyagi Prefecture have been experiencing severe isoyake[1], as shown in Fig. 1-1, which has led to a decrease in the number of fish and abalone becoming thin, causing a great loss to fishery production. One of the reasons for the occurrence of isoyake is thought to be the loss of algae beds due to the destruction of algae by the feeding of large numbers of sea urchins. To protect the algae and prevent the sea urchin population from continuing to grow[2], it is therefore necessary to remove the sea urchins as soon as possible. Currently, sea urchin removal is carried out by divers, but it is very hard and dangerous to work for long periods of time at depths of 10~20 m[3]. Therefore, it is necessary to develop a sea urchin removal ROV.

Our laboratory is currently developing a sea urchin removal large ROV[4] (Fig. 1-2) with a system that can recognize sea urchins and measure their size in real time and at high speed, where the specific specifications of the large ROV are shown in Table 1-1. As Fig. 1-3 shows the result of image recognition in Shizukawa Bay in December 2019[5].

1.2. Recycled sea urchins for aquaculture

The sea urchins recovered using the sea urchin removal large ROV described in 1.1 can be continued to be fed on aquaculture farms (Fig. 1-4), until they grow into edible fatty sea urchins, and establish a system to re-farm and sell these sea urchins, and the money from the sale can provide the necessary economic source for sustainable sea urchin aquaculture[6]. Currently farms have introduced system to manage information such as water temperature, salinity, and oxygen concentration, but the system is not very effective for sea urchins, as shown in Fig. 1-5.

In addition, sea urchins are highly likely to become sick due to the inability to ensure water quality in long-term aquaculture. Currently, sea urchin aquaculture farms use manual removal of these diseased sea urchins, which greatly restricts labor and does not guarantee that the situation inside the tank can be observed in any time. Therefore, there is a need to develop a sea urchin system using underwater stereo vision suitable for aquaculture farms.

1.3. Large sea urchin stereo recognition system

Since the sea urchin recognition system described in 1.1 (Fig. 1-6) uses hardware such as IMU, Laser, Raspberry Pi and so on, which is shown in detailed specifications in Table 1-2. Because the system is large and heavy, and the ROV used requires optical fiber cable, so it is not suitable for use on aquaculture farms. Considering the economic cost and the requirements of the farm environment, it is necessary to develop a low-cost, small, and simple structured sea urchin recognition ranging system.

1.4. Sea urchin aquaculture problems

Based on the problem in 1.2, we can conclude that the system to be developed for sea urchin aquaculture needs to have (1) low cost; (2) matching sea urchin size; (3) the ability to find diseased sea urchins. In this paper, we focus on (1) and (2). For (1), we take the approach of choosing as few hardware devices as possible; For (2), if the farmed sea urchins are not of the same size, it will happen that the larger size sea urchins will continue to get bigger while the smaller size sea urchins will get smaller in the aquaculture tank, which is highly susceptible to the phenomenon of cannibalism and slow growth. Moreover, due to the different sizes of sea urchins, the amount of bait fed cannot be unified, and it is very easy to have excess bait and thus deteriorate the water quality. For this reason, sea urchins are prone to be bald sea urchins (Fig. 1-7) and other specific diseases, so it is necessary to observe the condition of sea urchins in the tank in real time, sort out sea urchins of different sizes and feed them separately.

1.5. Sea urchin recognition system using underwater stereo vision

Based on (1) and (2) given in 1.4, there is a need to develop a sea urchin recognition system using underwater stereo vision for aquaculture farms by using a binocular camera to investigate the size of sea urchins. Therefore, this paper develops a simple camera structure that does not rely on other hardware, but only code to measure the sea urchins in the image by triangulation. The sea urchin recognition system using underwater stereo vision developed in this paper is based on a sea urchin recognition system[7] previously developed in the laboratory. After obtaining the specific image coordinates of the sea urchin, the similarity of the detection frames on the left and right images is calculated, and the datum points of the detection frames on the left and right images are input into the depth detection module. The depth detection module will perform two operations of stereo rectification and stereo matching on the left and right images to obtain the disparity map. Based on the coordinate values of the detected frames and the disparity map, we can get the distance between the detected sea urchin and the camera by calculating the formula, thus achieving the purpose of the research, as shown in Fig. 1-8, and the specific specifications of the sea urchin recognition system using underwater stereo vision are shown in Table 1-3. The specific application of the algorithm in this paper will be described in the following sections.

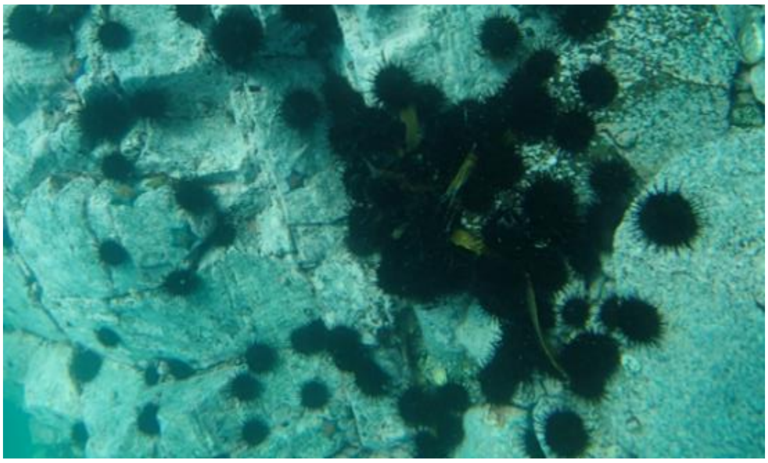


Fig. 1-1: Isoyake.

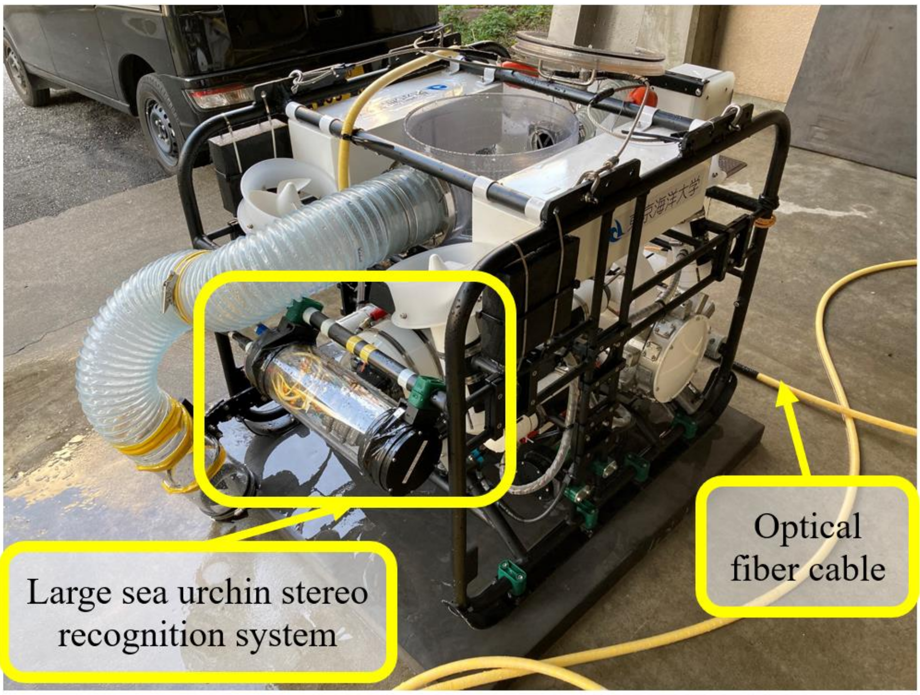


Fig. 1-2: Sea urchin removal large ROV.

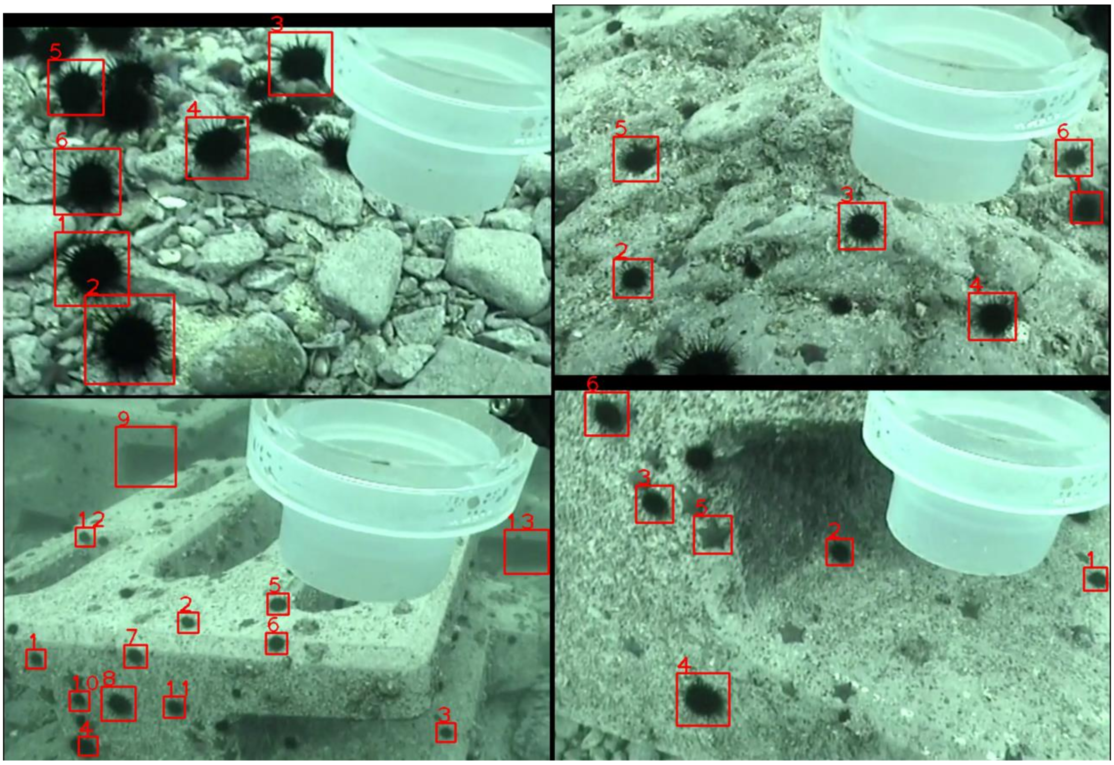


Fig. 1-3: Image recognition results for Shizukawa Bay, December 2019.

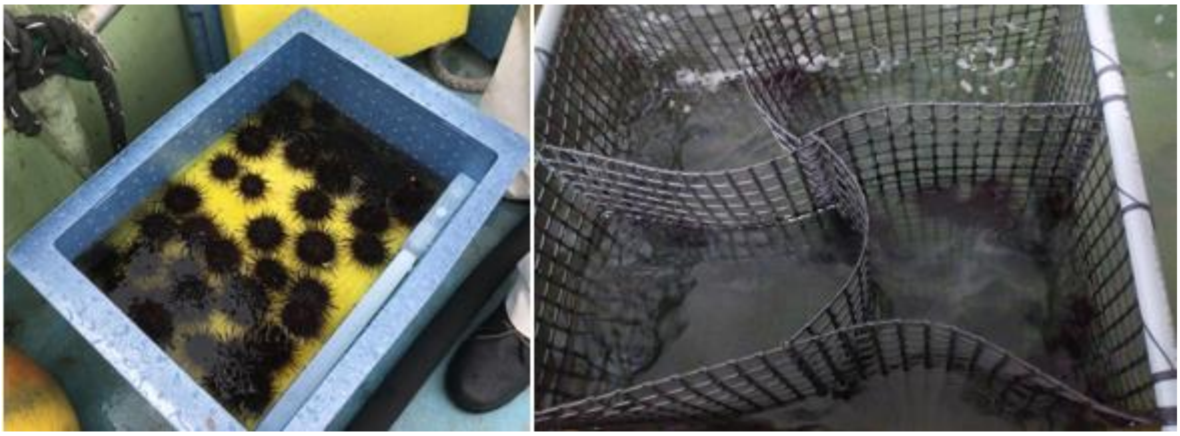


Fig. 1-4: Sea urchin aquaculture farm.

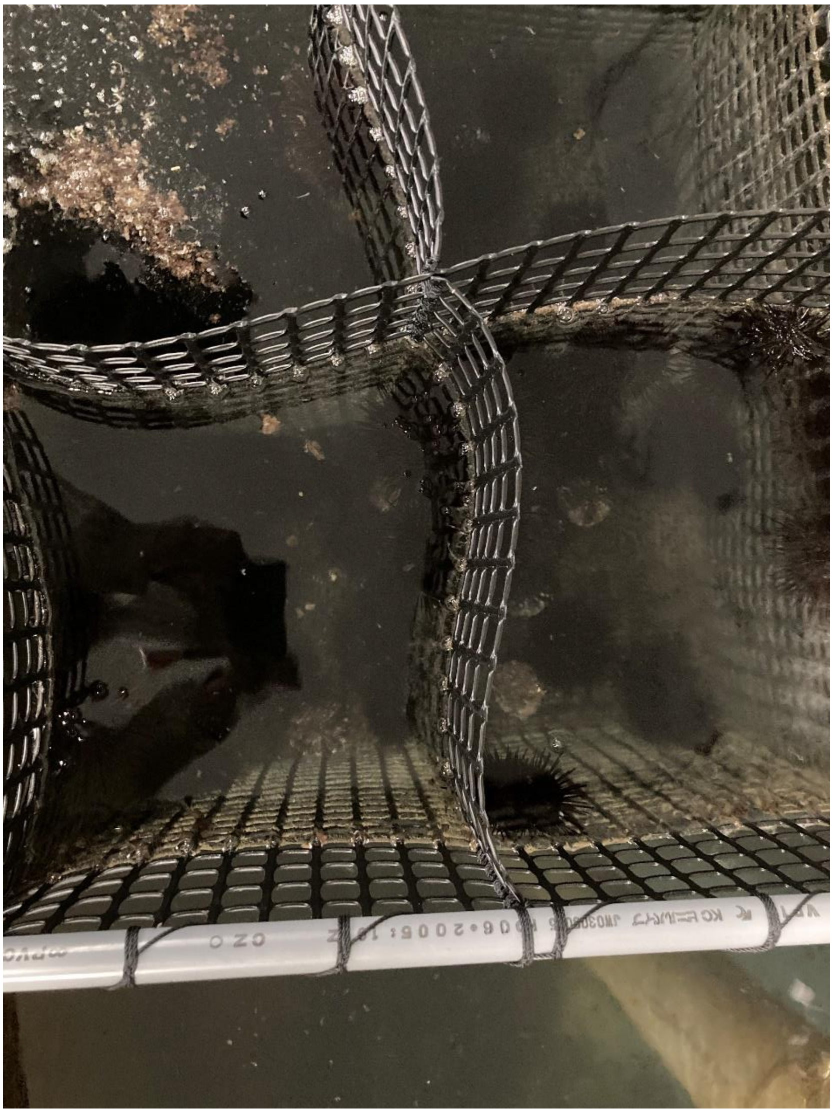


Fig. 1-5: Water quality conditions in sea urchin aquaculture farms.

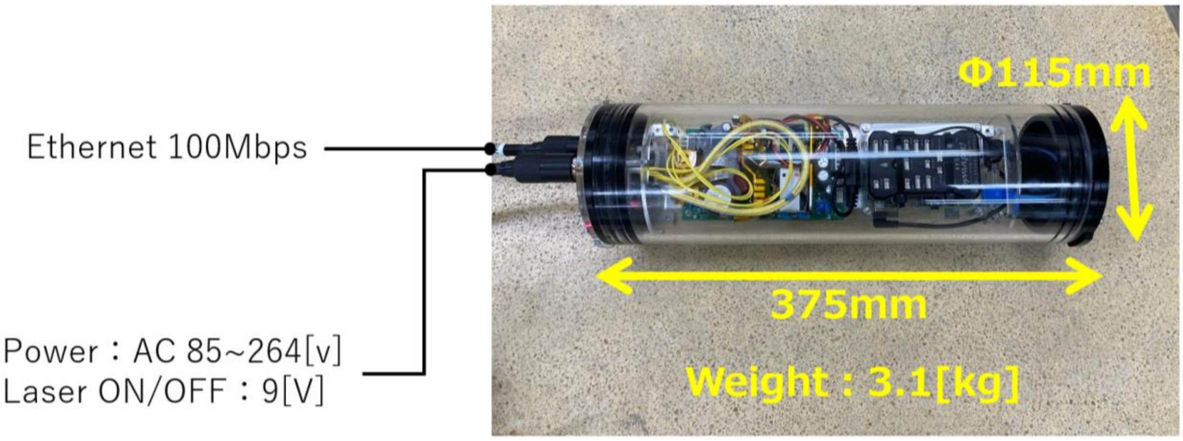


Fig. 1-6: Large sea urchin stereo recognition system.

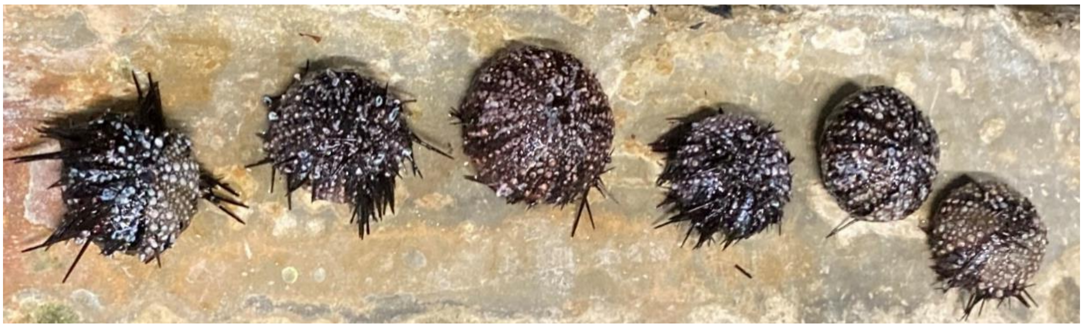


Fig. 1-7: Bald Sea urchin.

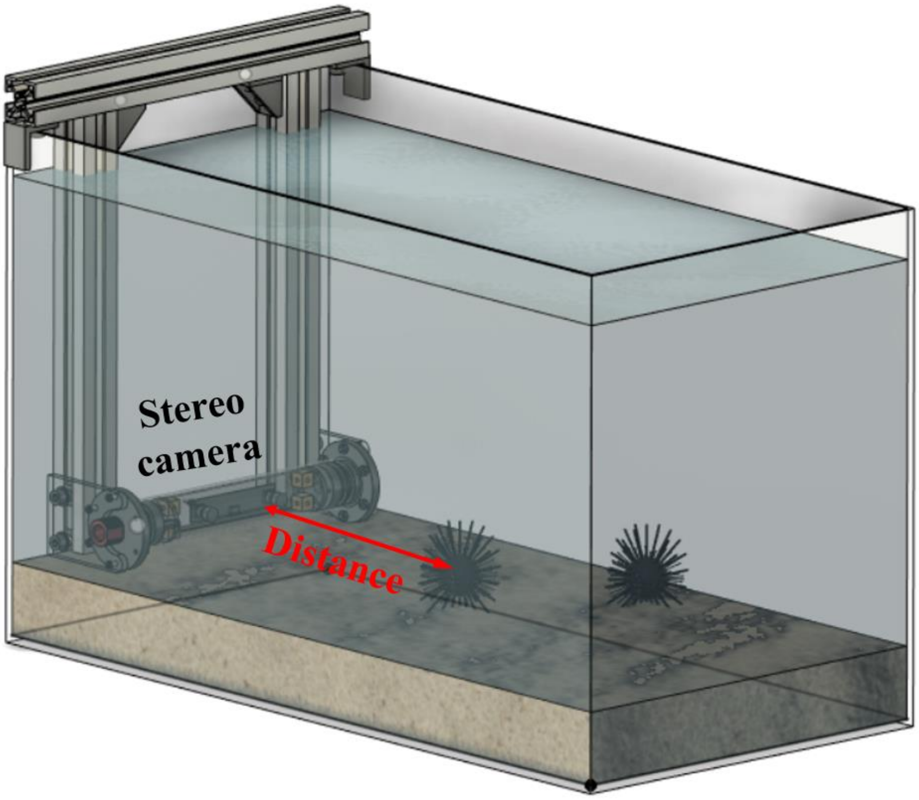


Fig. 1-8: Sea urchin recognition system using underwater stereo vision.

Table 1-1 Specifications of large ROV.

Item	Description
Shape	W635×L840×H490mm
Weight	45.4kg
Camera	NTSC
Binocular camera	ELP-1MP2CAM001-HOV90
Main thruster	Two horizons, two verticals(200W)
Sub-thruster	Two verticals(200W)
Robot hand	One unit
Sensor	IMU(Pixhawk), Laser, Raspberry Pi, AC - DC converter Depth gauge, Azimuth gauge, Altimeter

Table 1-2 Specifications of large sea urchin stereo recognition system.

Item	Description
Diameter	115[mm]
Length	375[mm]
Weight	3.1[kg]
Ethernet	100[Mbps]
Power	AC 85～264[V]
Laser ON/OFF	9[V]
Camera	ELP-1MP2CAM001-HOV90
Sensor	IMU(Pixhawk), Laser, Raspberry Pi, AC - DC converter

Table 1-3 Specifications of sea urchin recognition system using underwater stereo vision.

Item	Description
Diameter	55[mm]
Length	212[mm]
Weight	0.115[kg]
Power	DC 5[V]
Camera	ELP-1MP2CAM001-HOV90

2. Binocular calibration and image rectification

2.1. Camera calibration

In order to find the relationship between the pixel coordinate system and the world coordinate system, we need to calibrate the binocular camera. Fig. 2-1 shows the binocular camera ranging model, where the coordinate system of the x_w, y_w, z_w axes is the world coordinate system, and its position in the stereoscopic space can be placed at will. The coordinate system of the *x*, *y* axes is the image coordinate system, the coordinate system formed by the x_c, y_c, z_c axes are the camera coordinate system, and the coordinate system of the u, v axes component is the pixel coordinate system. These coordinate systems together constitute the distance measurement system.

As shown in Fig. 2-1, point *P* is a point in the target object, *p_{l,r}* are *P* a point projected onto the imaging plane. The intersection of the camera optical axis with the image plane (generally located at the center of the image plane, also called the principal point of the image) is defined as the origin *O_{Il,Ir}* of camera coordinate system, and *O_{pl,pr}* is origin of the pixel coordinate, with the *x_{l,r}* -axes is parallel to the *u_{l,r}*-axes, and the *y_{l,r}*-axes are parallel to the *v_{l,r}*-axes. Assuming (u_0, v_0) and (u'_0, v'_0) as the origin of the pixel coordinate system, without considering the distortion, and *dx* and *dy* represent the physical size of each pixel on the horizontal and vertical axes, respectively. The coordinate transform model is given by Fig. 2-2, where the camera intrinsic matrix(*M_{l,r}*,*D_{l,r}*) and external matrix(*R*, *T*) can be obtained by calibration experiments.

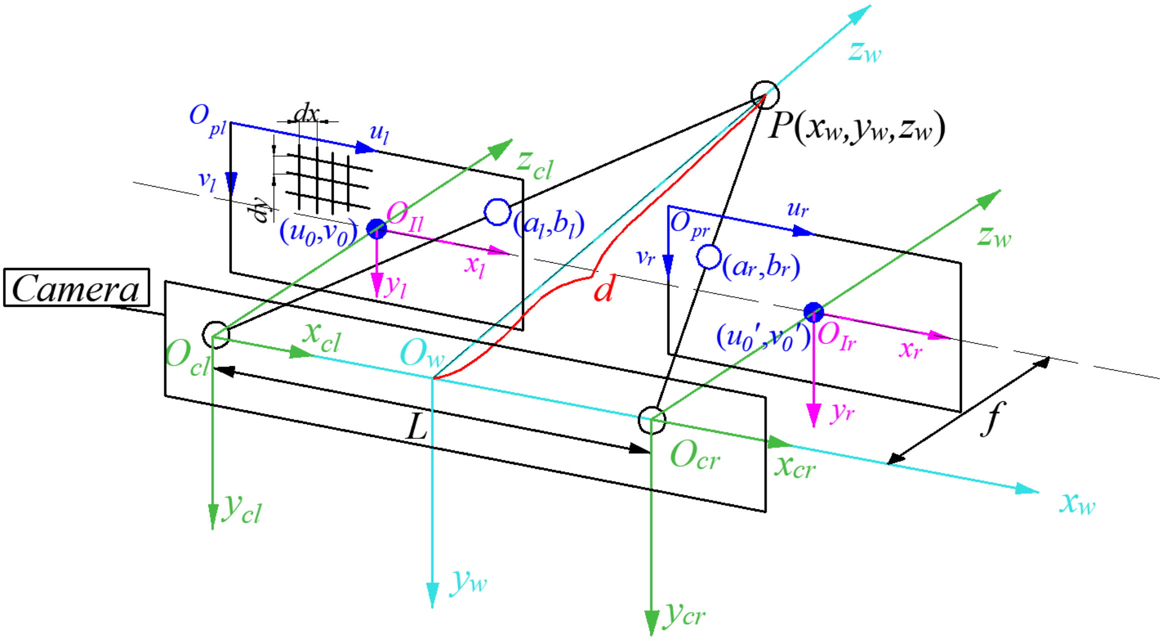


Fig. 2-1: Binocular camera ranging model

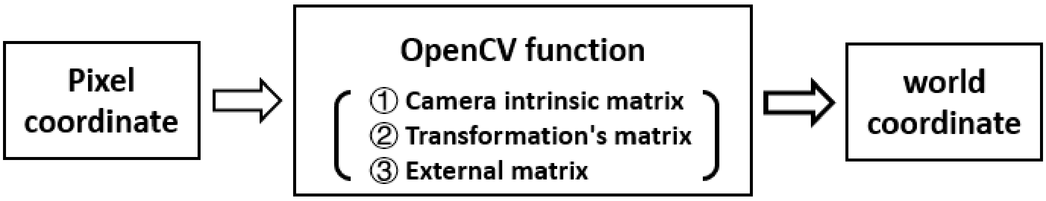


Fig. 2-2: Binocular range coordinate transform model.

2.2. Geometric relationship of coordinate systems

According to Fig. 2-1, we need to correspond the points in pixel coordinates to the actual points in the world coordinate system. By using Zhang's algorithm and OpenCV, the real distance can be obtained.

2.2.1. Conversion between image coordinate system and pixel coordinate system

Firstly, we need to perform the conversion from pixel coordinates to image coordinates. The pixel coordinate system is expressed in pixel units and the image coordinate system is expressed in physical units. As shown in Fig. 2-3, we can see that the horizontal and vertical axes between the two coordinate systems are parallel to each other and shifted by a certain distance. *u* and *v* of the pixel coordinates correspond to the rows and columns of the image； while the image coordinate system represents the specific physical length，where *dx* is the unit length of the horizontal *x*-axis and *dy* is the unit length of the vertical *y*-axis.

Where the blue coordinate system represents the pixel coordinate system, and the pink coordinate system represents the image coordinate system. The coordinate system with *u* and *v* axes is the pixel coordinate system with coordinate origin (0,0); the coordinate system with *x* and *y* axes is the image digital coordinate system with coordinate origin represented by (u_0, v_0). The relationship[8] between these two coordinate systems is shown in Eq. (2-1).

$$\begin{cases} u = u_0 + \dfrac{x}{dx} \\ v = v_0 + \dfrac{y}{dy} \end{cases} \tag{2-1}$$

For ease of later calculation, Eq. (2-1) is converted to homogeneous coordinates, as shown in Eq. (2-2).

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \dfrac{1}{dx} & 0 & u_0 \\ 0 & \dfrac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2-2}$$

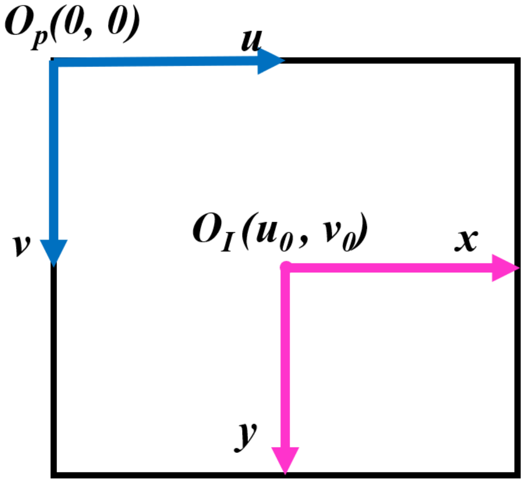


Fig. 2-3: Position relationship between pixel coordinate system and image coordinates.

2.2.2. Conversion between camera coordinate system and image coordinate system

Secondly, we need to perform the conversion from image coordinates to camera coordinates. As shown in Fig. 2-4, the origin of the camera coordinate system is located at the center of the camera, that is the optical center point. The axis of the camera coordinate system is perpendicular to the camera lens, that is the image plane. The relationship between the camera coordinate system and the image coordinate system is shown in Eq. (2-3).

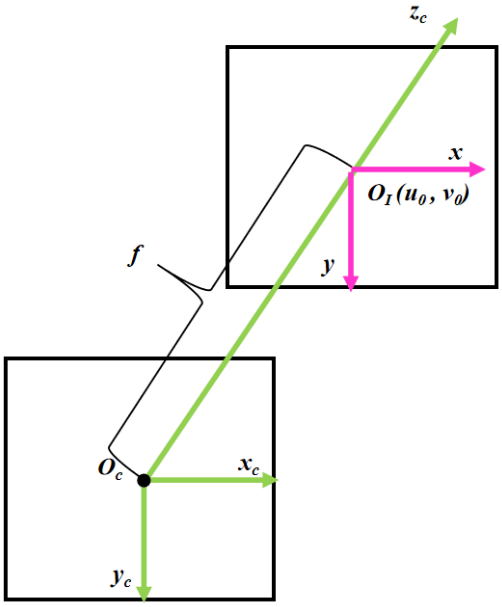


Fig. 2-4: Position relationship between image coordinate system and camera coordinates.

$$\begin{cases} x = \dfrac{f}{z_c} \cdot x_c \\ y = \dfrac{f}{z_c} \cdot y_c \end{cases} \tag{2-3}$$

The conversion to the homogeneous coordinates is shown in Eq. (2-5).

$$z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{2-4}$$

2.2.3. Conversion between world coordinate system and camera coordinate system

Thirdly, we need to perform a conversion from the camera coordinate system to the world coordinate system. The world coordinate system consists of the x_w, y_w, and z_w axes, while the camera coordinate system can be overlapped with the world coordinate system by rotations and translations, and their conversion relationship is shown in Eq. (2-5).

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{2-5}$$

where R represents the rotation matrix(3×3) and T represents the translation vector(3×1). R and T denote the external parameters of the position of the camera in relation to the image plane.

2.2.4. Conversion between world coordinate system and pixel coordinate system

Finally, we need to perform a direct conversion from pixel coordinates to world coordinates, as shown in Eq. (2-6).

$$z_c\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} \partial_x & 0 & u_0 & 0 \\ 0 & \partial_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{2-6}$$

To simplify the above equations, we can use $Q_a = M = \begin{bmatrix} \partial_x & 0 & u_0 & 0 \\ 0 & \partial_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ and $Q_b = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix}$, and the projection matrix Q can relate the pixel coordinate and the world coordinate, where $Q=Q_a \times Q_b$. Using the projection matrix Q, we can obtain information about of the world coordinate system based on the known values of the pixel coordinate of the tested point P.

2.3. Distortion coefficient

Due to the lens material and process, lens distortion is difficult to avoid in camera shooting. Lens distortion mainly includes radial distortion and tangential distortion[9]. Which radial distortion phenomenon as shown in Fig. 2-5.

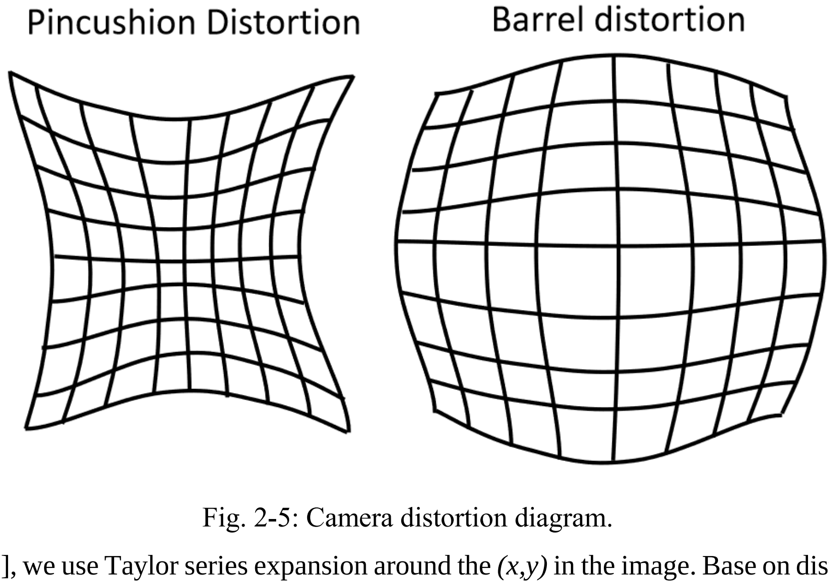


Fig. 2-5: Camera distortion diagram.

According to [9], we use Taylor series expansion around the *(x,y)* in the image. Base on distortion parameters *D_{l,r}=[*k_1*,*k_2*,*p_1*,*p_2*,*k_3*]*, the radial distortion can be expressed by Eq. (2-7), which *(x,y)* is original image's point and *(x',y')* is rectified image's point.

$$\begin{cases} x^{'} = x\left(1+k_1r^2+k_2r^4+k_3r^6\right) \\ y^{'} = y\left(1+k_1r^2+k_2r^4+k_3r^6\right) \end{cases} \tag{2-7}$$

The tangential distortion is expressed in Eq. (2-8).

$$\begin{cases} x^{'} = x+\left[2p_1y+p_2\left(r^2+2x^2\right)\right] \\ y^{'} = y+\left[2p_2x+p_1\left(r^2+2y^2\right)\right] \end{cases} \tag{2-8}$$

where k_1, k_2, k_3 represent the radial distortion coefficients and p_1, p_2 are tangential distortion. These parameters can be obtained through calibration experiments.

2.4. Selection of calibration algorithm

Currently, there are two methods of camera calibration. One is the traditional calibration method of the camera to have a reference; Another is the self-calibration method that does not require a reference. The traditional calibration method requires a stricter reference, but the accuracy of the parameters obtained is better. Zhang's algorithm[10] is between the traditional calibration method and the self-calibration method, which solves the shortcomings of the traditional calibration method that requires strict reference. At the same time, it improves the accuracy and is easy to operate.

Zhang's algorithm uses chessboard pictures for calibration, it can obtain the correspondence between references and images, which not only provides more accurate calibration results, but also better real-time performance. In addition, we can use two ordinary cameras to achieve the experimental purpose with relatively low cost.

At present camera calibration can be implemented by means of MATLAB and OPENCV. For the sake of platform uniformity and ease of parameter transfer, the OpenCV-Python approach was chosen for the calibration of the camera in this paper. Since the rotation matrix R and translation vector T obtained by the OpenCV library cv2.stereoCalibrate() have too large errors, a monocular calibration followed by binocular calibration was chosen for the calibration of the binocular camera.

2.5. Calibration experiment

In this paper, the ELP-1MP2CAM001-HOV90 dual-lens camera module as shown in Fig. 2-6. The specifications of the camera module are shown in the Table 2-1. The binocular camera is calibrated according to Zhang's algorithm using OpenCV-Python. The calibration pattern used for the calibration experiment is a black and white chessboard grid, each square of which is 25 mm in size and has 9×6 corner points. The reason for choosing the chessboard grid for the experiment is that the chessboard grid has the characteristics of a horizontal and vertical grids, and the corner points are easy to identify. The principle of binocular camera calibration is to use two cameras that are ideally parallel and have identical specifications for calibration experiments, and to find the internal and external parameters of the binocular camera through OpenCV calibration.

Through the continuous calibration experiments on the binocular camera, three sets of parameters with better calibration results are obtained, respectively denoted as A, B, and C. The graphs of calibration results are shown in Fig. 2-9.

Zhang's algorithm process: Firstly, images are taken by binocular cameras and took at least 80 sets of image pairs are obtained, as shown in Fig. 2-7. Then the corner points of each image are extracted, the extracted corner point position information is judged, and these corner points are plotted, as shown in Fig. 2-8. If some of these corner points are judged to be wrong corner points then the corner point extraction is performed again, and the correct corner point coordinates will be calculated for the next calibration step. The flow chart of Zhang's calibration is shown in Fig. 2-10.

Table 2-1: Binocular camera specifications.

Model	ELP-1MP2CAM001-HOV90
Lens size	1/4 inch
Pixel size	3.0 um×3.0 um
Image area	3888um×2430 um
Max resolution	1280(H)×720(V)
Compression format	MJPEG / YUV2(YUYV)
S/N ration	40 dB

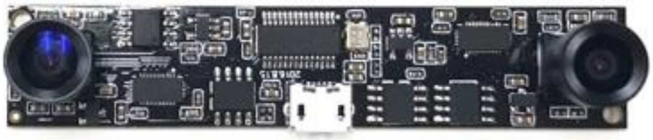


Fig. 2-6: ELP-1MP2CAM001-HOV90 dual-lens camera module

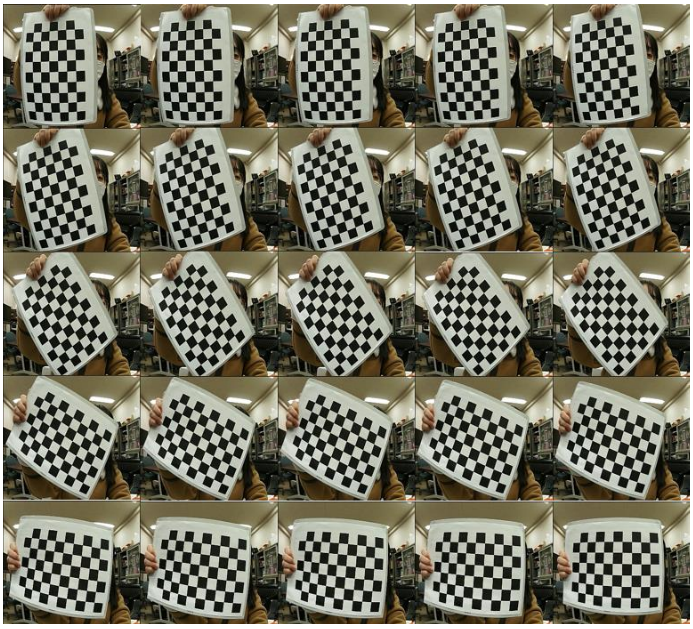


Fig. 2-7: Binocular camera calibration images.

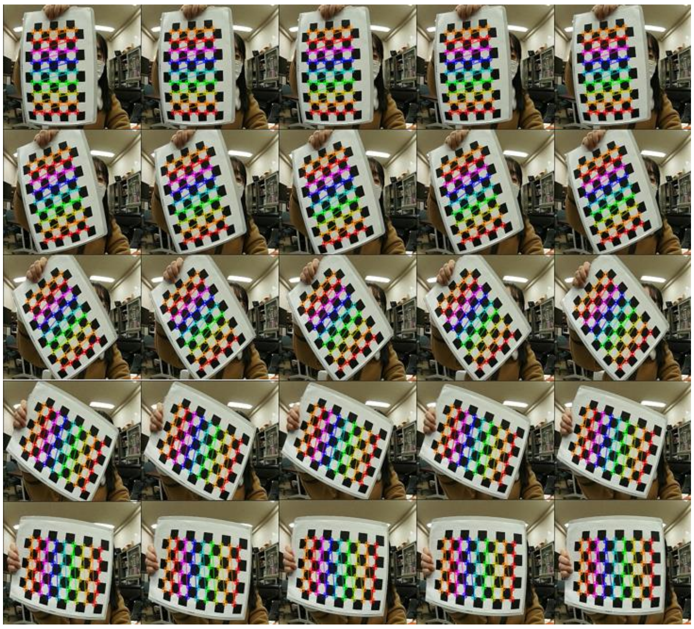


Fig. 2-8: Binocular camera calibration result images.

A

$$M_l = \begin{bmatrix} 237.528275 & 0 & 156.407314 \\ 0 & 238.041493 & 112.673873 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_r = \begin{bmatrix} 225.420156 & 0 & 151.819540 \\ 0 & 226.342191 & 114.235772 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D_l = [-0.404995 \quad 0.173975 \quad -0.007028 \quad -0.000219 \quad 0.019297]$$

$$D_r = [-0.365212 \quad 0.106109 \quad -0.009129 \quad -0.001468 \quad 0.144811]$$

$$R = \begin{bmatrix} 0.999964 & -0.001158 & -0.0083644 \\ 0.001051 & 0.999917 & 0.012856 \\ 0.008379 & 0.012847 & 0.999882 \end{bmatrix}$$

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} 61.765999 \\ -0.696587 \\ -13.849930 \end{bmatrix}$$

B

$$M_l = \begin{bmatrix} 238.003920 & 0 & 156.488966 \\ 0 & 238.060968 & 110.441979 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_r = \begin{bmatrix} 219.429299 & 0 & 150.532048 \\ 0 & 219.944700 & 116.770279 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D_l = [-0.419733 \quad 0.142441 \quad 0.000991 \quad -0.001641 \quad 0.059386]$$

$$D_r = [-0.359667 \quad 0.144886 \quad -0.0069 \quad -0.002087 \quad 0.019845]$$

$$R = \begin{bmatrix} 0.999991 & -0.001512 & -0.003916 \\ 0.001426 & 0.9999761 & -0.021823 \\ 0.003948 & 0.021817 & 0.9999754 \end{bmatrix}$$

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} 62.044231 \\ -3.550818 \\ -20.614803 \end{bmatrix}$$

C

$$M_l = \begin{bmatrix} 231.944174 & 0 & 158.585784 \\ 0 & 231.964340 & 111.247045 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_r = \begin{bmatrix} 220.544354 & 0 & 152.107704 \\ 0 & 221.214708 & 120.118674 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D_l = [-0.412504 \quad 0.203156 \quad -0.002040 \quad -0.004228 \quad -0.046696]$$

$$D_r = [-0.376732 \quad 0.194905 \quad -0.011775 \quad -0.005215 \quad 0.034463]$$

$$R = \begin{bmatrix} 0.999985 & -0.005202 & -0.001783 \\ 0.005140 & 0.999442 & -0.033002 \\ 0.001954 & 0.032992 & 0.999454 \end{bmatrix}$$

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} 62.337653 \\ -3.726065 \\ -14.226682 \end{bmatrix}$$

Fig. 2-9: Binocular camera calibration result parameters.

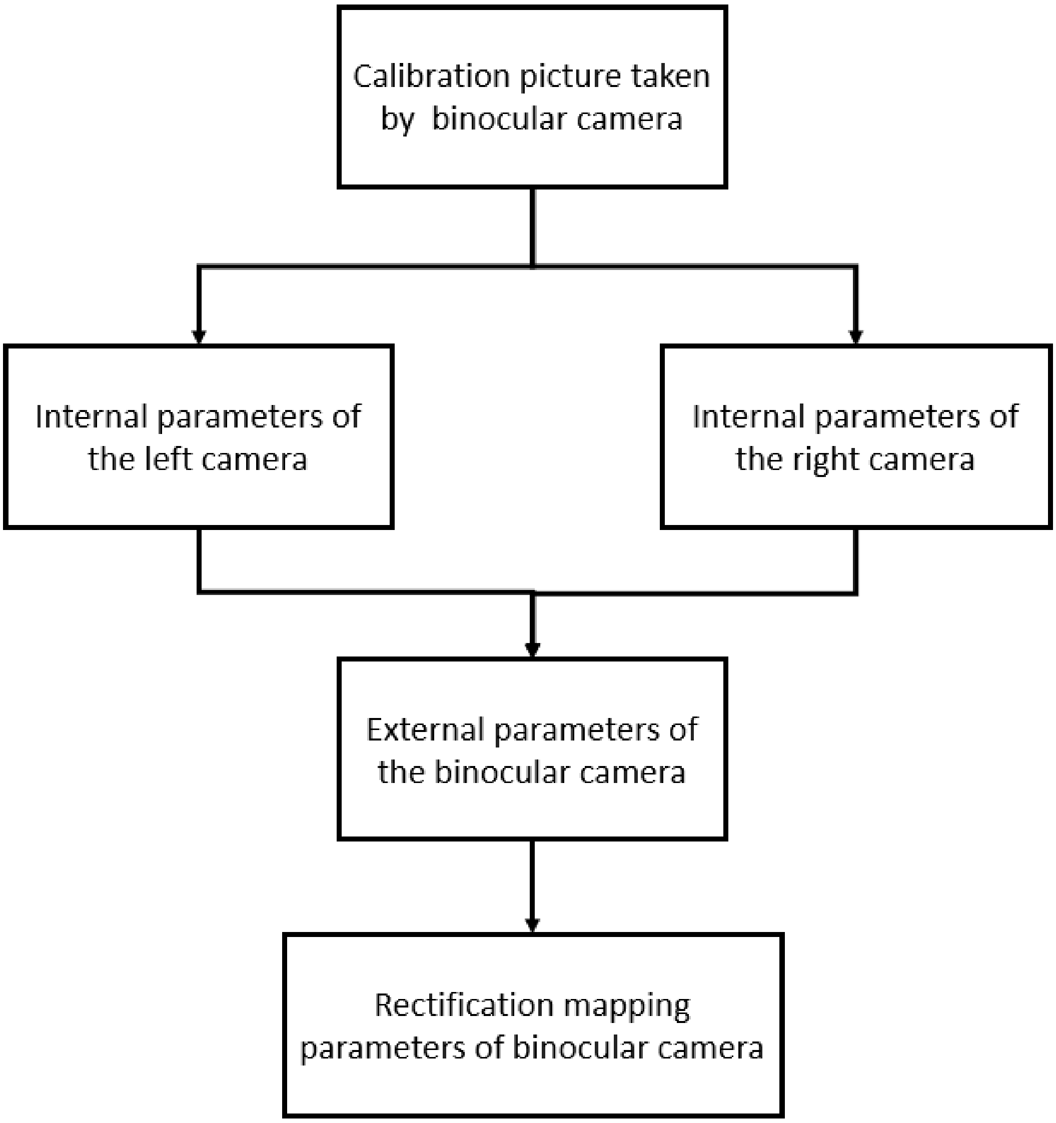

Calibration picture taken by binocular camera
Internal parameters of the left camera
Internal parameters of the right camera
External parameters of the binocular camera
Rectification mapping parameters of binocular camera

Fig. 2-10: Binocular calibration flow chart.

3. Image processing and sea urchin recognition

This chapter mainly introduces the image processing and sea urchin recognition of sea urchin system using underwater stereo vision. The main purpose of image processing is to ensure that the quality of the images is not affected by noise. Sea urchin recognition uses sea urchin classifier previously developed in the laboratory to recognize the sea urchins that appear in the images and thus obtain the coordinate values of the sea urchins in the image coordinate system.

3.1. Image processing

Image processing is mainly aimed at eliminating irrelevant information from images, recovering useful real information, enhancing detectability of relevant information and maximizing data simplification. Thus, the possibilities of sea urchin recognition, stereo matching and depth detection are improved.

3.1.1. Image grayscale processing

The camera captures the raw image in color, but color images contain the full information of the image, computer processing of color images is much less effective than grayscale images in terms of time and effectiveness. In binocular camera ranging, grayscale images are sufficient to represent the complete information of the image, so it is necessary to process the image in grayscale.

Grayscale images are stored using *8* nonlinear scales per pixel, so that there is $8\times8\times8=255$ different grayscale values, which is enough to represent the complete information of the image, making the image undistorted and easier to transmit, and the memory for storing images after grayscale processing is greatly reduced. The specific process is shown in Fig. 3-1.

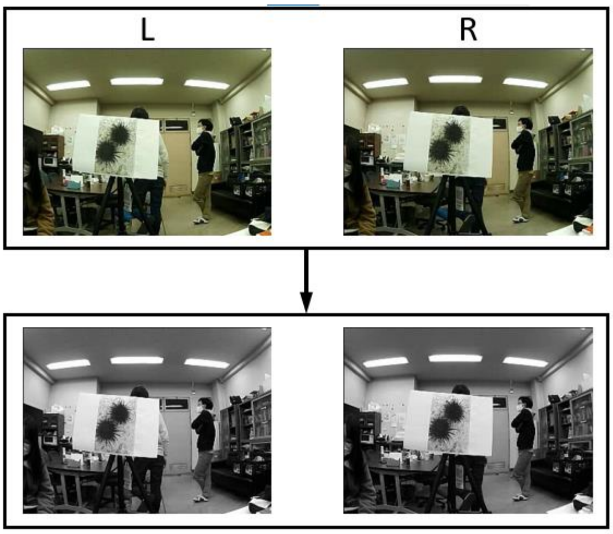


Fig. 3-1: Image graying process diagram.

3.1.2. Image enhancement

To enhance the contrast between the image sea urchin and the background, we use the histogram equalization to enhance the dynamic range of the pixel gray values by distributing the pixel values evenly over the entire gray range (0~255) of the gray image. In this paper, we use the cv2.equlizeHist() in the OpenCV library to enhance the image. The specific process is shown in Fig. 3-2.

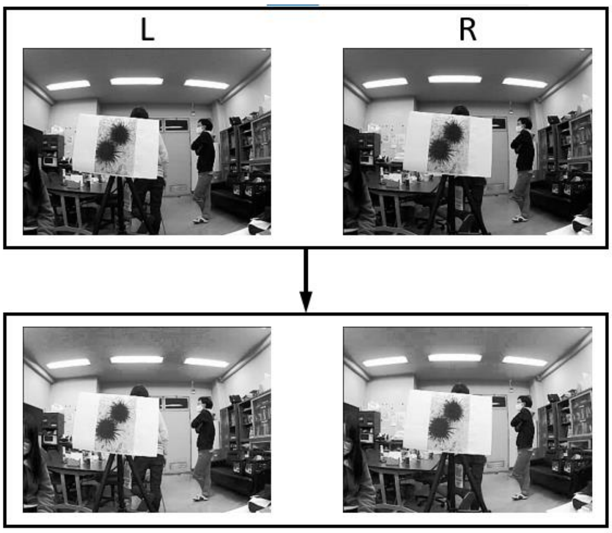


Fig. 3-2: Image enhancement process diagram.

3.1.3. Image filtering

In camera shooting, a variety of noises are generated due to the image transmission process, and these noises disrupt the observable information of the image. So, it is necessary to perform image filtering on the image. The current filters are [11]: mean filter, median filter, Gaussian filter and Bilateral filter.

(1) Mean filter (blur)

Mean filter is a smooth linear spatial filter that mainly uses the average value of the pixels around a pixel to achieve the effect of smoothing noise. Mean filter is a low-pass filter, which means that the mean value in the domain is assigned to the central element.

Mean filter is used to reduce noise and is mainly used to remove irrelevant details from the image, where irrelevant is the area of pixels that are smaller compared to the filter's template. Blurred images are used to get a rough description of the object of interest, so that the grayscale of those smaller objects is blended with the background and larger objects become speckle-like and easy to detect. The size of the template is determined by the size of those objects that will blend into the background.

(2) Median filter

Median filter is a nonlinear filter that is often used to eliminate salt-and-pepper noise in images. Unlike low-pass filtering, median filtering is good for preserving the sharpness of edges, but it washes away the texture in the uniform media area.

Salt-and-pepper noise is the bright and dark noise in black and white generated by the image sensor, transmission channel, decoding process and so on. Salt-and-pepper noise refers to two kinds of noise, one is salt noise whose salt = white (255), and the other is pepper noise whose pepper = black (0). The former is a high gray noise, the latter belongs to the low gray noise. Generally, the two kinds of noises appear at the same time, presented in the image is black and white miscellaneous color. For color images, it is presented as 255 and 0 that appear randomly in the three channels of a single pixel BGR.

(3) Gaussian filter

Gaussian filter is one of the linear filters. Gaussian filter is used for smoothing images, or image blurring process, so Gaussian filter is a low-pass filter. Its widely used in the noise reduction process of image processing, especially on images contaminated by Gaussian noise.

The value of each pixel point on image is obtained by a weighted average of its own value and the values of other pixel points in the domain. Gaussian filter is implemented by scanning each pixel point in the image with a kernel (convolution kernel), multiplying each pixel value in the domain with the weight value at the corresponding position and summing them. Mathematically, the process of Gaussian filtering is a convolution operation of the image with the Gauss normal distribution.

(4) Bilateral Filter

Bilateral filter is a nonlinear filtering that combines the approach of image spatial proximity and pixel value similarity. In filtering, this filtering method considers both spatial proximity information and color similarity information, removing noise and smoothing the image while achieving edge preservation.

In general, Bilateral filter uses a combination of two Gaussian filters. One is responsible for calculating the weight of spatial proximity, which is the principle of the usual Gaussian filter. The other one is responsible for calculating the weights of pixel value similarity.

3.2. Sea urchin recognition

The image recognition system in this paper focuses on creating classifiers rapidly building classifiers with the available GPU performance, using the pattern matching method (Viola-Jones[12]) to produce a recognition classifier, and to generate stronger classifier to detect objects from many weaker classifiers. To create the classifier, positive samples (sea urchins) and negative samples (not sea urchins) are required, and then a list of image labels and vector files (automatically generated via opencv-createsamples.exe) are created for the positive and negative samples, respectively, as shown in Fig. 3-3. In this paper, the vector files generated above are used to recognize and extract features from the images using Haar-like features. The sea urchin in the image is recognized according to the classifier and the pixel coordinates of the sea urchin detection frames are obtained, providing the target object coordinates for subsequent stereo matching. Since we perform binocular ranging for specific object (sea urchin), this system ranges the sea urchin patches detected by the left and right cameras at the same time and calculates the similarity of the two patches to get the sea urchin we need.

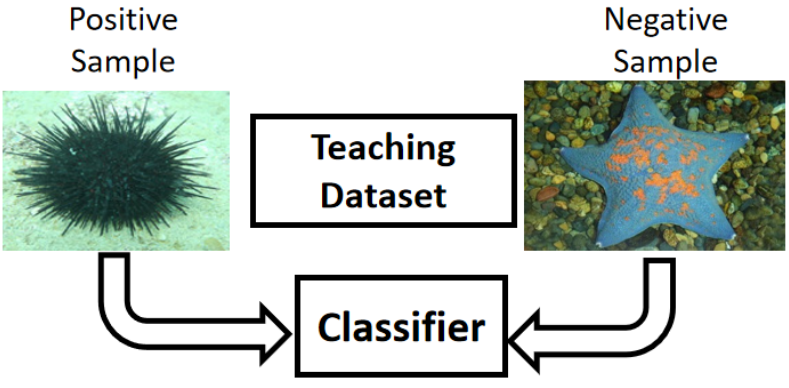


Fig. 3-3: Sea urchin classifier generation schematic.

4. Depth detection system

4.1. Image rectification

4.1.1. Bouguet's algorithm

In this paper, we use the Bouguet's algorithm, which aims to minimize the reprojection variation in each of the two images (thus minimizing the result of reprojection distortion) while maximizing the common field of image.

To minimize the reprojection distortion of the image, we can divide the rotation matrix *R* into two parts r_r and r_l that make the two rotation matrices of the left and right cameras each rotate half a turn, thus their principal rays will end up parallel to the vector sum where their original principal rays are pointing. But this only puts the two cameras into coplanar alignment, not line alignment. We can translate the epipole $\overrightarrow{e_1}$ of the left camera to infinity, with the principal point (c_x,c_y) as the origin of the left image, and the translation vector between the projection centers of the two cameras along the direction of the epipolar line can be shown in Eq. (4-1) below[9].

$$\overrightarrow{e_1} = \frac{\vec{T}}{\left\|\vec{T}\right\|} \tag{4-1}$$

We need to make a vector $\overrightarrow{e_2}$ orthogonal to $\overrightarrow{e_1}$. We can choose a direction orthogonal to the principal ray (which tends to be along the image plane) as the direction of $\overrightarrow{e_2}$, and get the direction of $\overrightarrow{e_2}$ by making the cross-product of $\overrightarrow{e_1}$ with the direction of the principal ray, and then obtain another unit vector $\overrightarrow{e_3}$ by normalizing.

There is always a third vector $\overrightarrow{e_3}$, that is orthogonal to $\overrightarrow{e_1}$ and $\overrightarrow{e_2}$ by using the cross-product operation, as show in Eq. (4-2).

$$\overrightarrow{e_3} = \overrightarrow{e_1} \times \overrightarrow{e_2} \tag{4-2}$$

The matrix R_{rect} that transforms the epipole in the left camera to infinity is show in Eq. (4-3).

$$R_{rect} = \begin{bmatrix} \overrightarrow{e_1}^T \\ \overrightarrow{e_2}^T \\ \overrightarrow{e_3}^T \end{bmatrix} \tag{4-3}$$

The matrix R_{rect} rotates the left camera around the projection center so that the epipolar lines are horizontal and the epipoles are at infinity. The row alignment of two of the cameras can be achieved according to Eq. (4-4) and Eq. (4-5)

$$R_l = R_{rect} \cdot r_l \tag{4-4}$$

$$R_r = R_{rect} \cdot r_r \tag{4-5}$$

We will calculate the rectified left and right camera matrices $M_{rect,l}$ and $M_{rect,r}$, and substitute them into the projection matrices $P_l^{'}$ and $P_r^{'}$ as follows Eq. (4-6) and Eq. (4-7).

$$P_l = M_{rect,l} \cdot P_l^{'} = \begin{vmatrix} f_{rect,l} & \alpha_l & c_{x,l} \\ 0 & f_{y,l} & c_{y,l} \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{vmatrix} \tag{4-6}$$

$$P_r = M_{rect,r} \cdot P_r^{'} = \begin{vmatrix} f_{rect,r} & \alpha_r & c_{x,r} \\ 0 & f_{y,r} & c_{y,r} \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{vmatrix} \tag{4-7}$$

Using Eq. (4-8), the projection matrix *P* converts a three-dimensional point in a homogeneous coordinate system into a two-dimensional point in homogeneous coordinates.

$$P \cdot \begin{vmatrix} X \\ Y \\ Z \\ 1 \end{vmatrix} = \begin{vmatrix} x \\ y \\ w \end{vmatrix} \tag{4-8}$$

The image coordinates can be calculated according to (*x*, *y*) = (*x/w*, *y/w*). Given the image coordinates and the camera intrinsic matrices, the two-dimensional points can be reprojected into three-dimensional. Where the projection matrix *Q* is shown in Eq. (4-9), where *P* is the ideal data of *Q*.

$$Q = \begin{vmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -\dfrac{1}{T_x} & \dfrac{c_x - c_x^{'}}{T_x} \end{vmatrix} \tag{4-9}$$

$c_x^{'}$ is the parameter from the left image, and it is the principal point of the x-coordinate axis in the right image. If the principal rays of the left and right cameras intersect at the point of infinity ($c_x = c_x^{'}$), so the term in the lower right corner of the projection matrix *Q* is 0. By using Eq. (4-10), we can project the measured point into three-dimensional space with a two-dimensional homogeneous point and its related disparity *d*.

$$Q \begin{vmatrix} x \\ y \\ d \\ 1 \end{vmatrix} = \begin{vmatrix} X \\ Y \\ Z \\ W \end{vmatrix} \tag{4-10}$$

The three-dimensional coordinates are $\left(\dfrac{X}{W}, \dfrac{Y}{W}, \dfrac{Z}{W} \right)$.

To maximize the overlapping image area, the rotated image can be set with a new image center and a new image boundary, setting the unified camera center and the common maximum height and width of the two image regions to the center's stereo viewing plane. Fig. 4-1 shows a schematic diagram of the Bouguet’s rectification principle.

In this paper, we use the cv2.stereoRectify() function in the OpenCV library file to compute the rectification and disparity maps we need to extract depth information from the stereo images of the binocular camera.

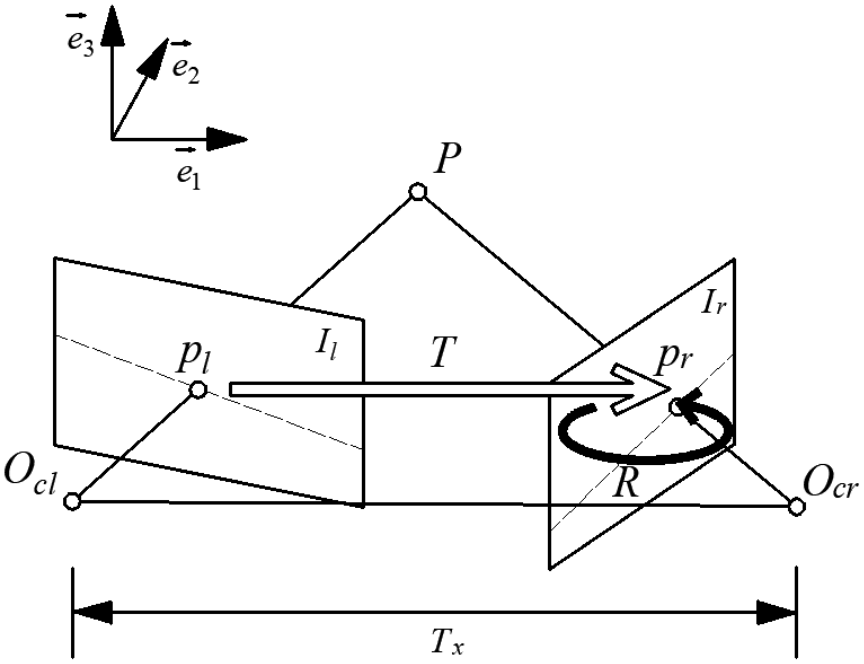


Fig. 4-1 shows a schematic diagram of the Bouguet's rectification principle.

4.1.2. Rectification Mapping

In this paper, we use the cv2.initUndistortRectifyMap() function to pre-calculate the rectification maps for the left and right images. Here only the pixel positions from the source image to the target image are calculated. For each integer pixel position in the target image, the corresponding pixel position of the source image is found by finding the floating-point coordinates on the source image and bilinear interpolation using the integer values of the surrounding source pixels.

For each integer pixel on the rectified image, the corresponding coordinates can be found in the undistorted image. In this paper, we use the reverse thinking of the rectified image of to find the corresponding coordinate values of the target image by the coordinate values of the raw image. The pixel values on the floating point coordinates are obtained by interpolating the neighboring integer pixel positions on the raw image, and this value will be assigned to the corresponding interpolation position of the rectified image, followed by cropping of the rectified image to increase the superimposed area between the left and right images[9]. The whole rectification process is shown in Fig. 4-2.

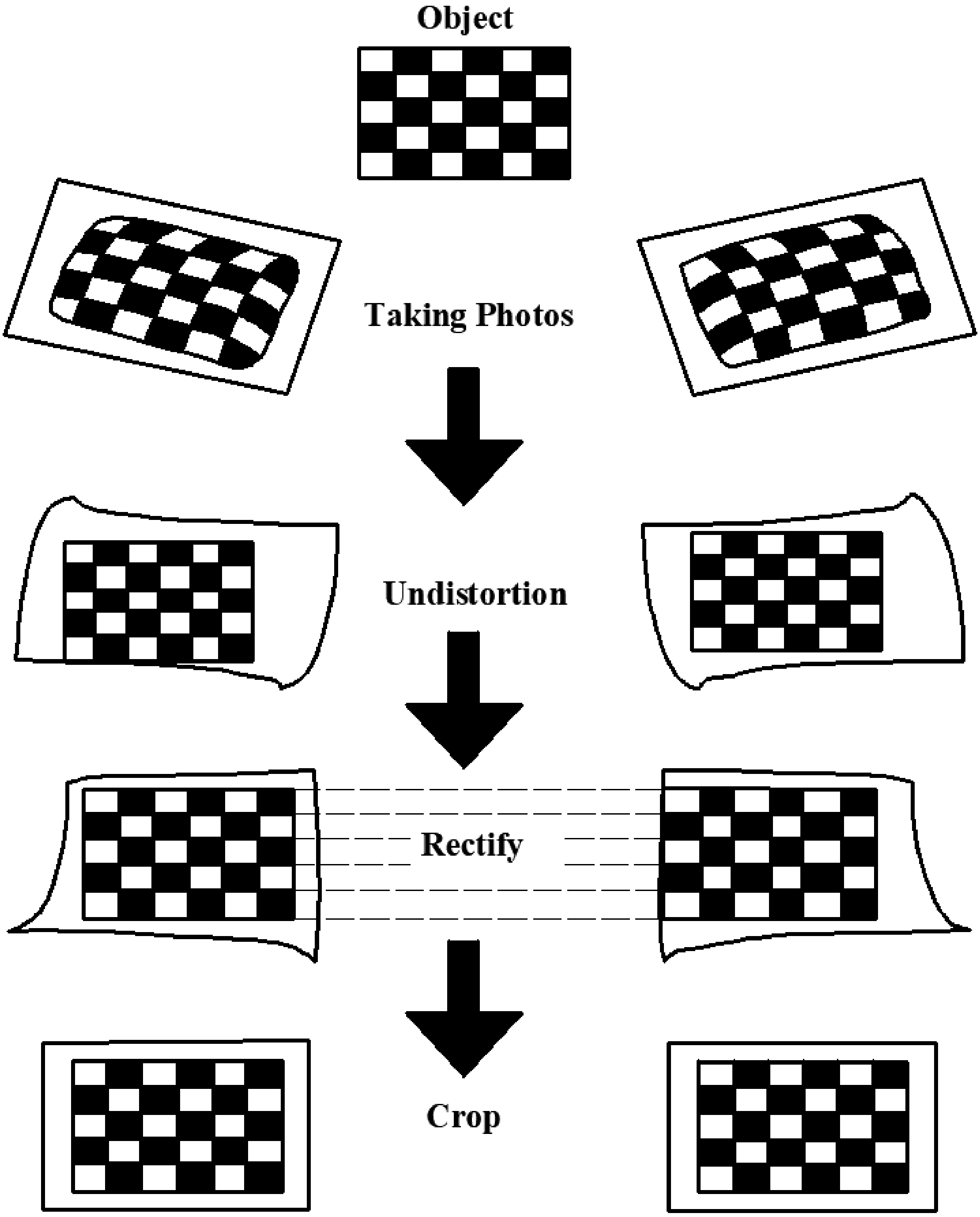


Fig. 4-2: Stereo rectification flow chart.

We rectified the images obtained from the binocular camera, and we use the cv2.remap() function to specify the rectified image position for each pixel in this image using interpolation. The Fig. 4-3 shows the process of image interpolation, where u and v represent the pixel coordinate values. The real effect is shown in Fig. 4-4.

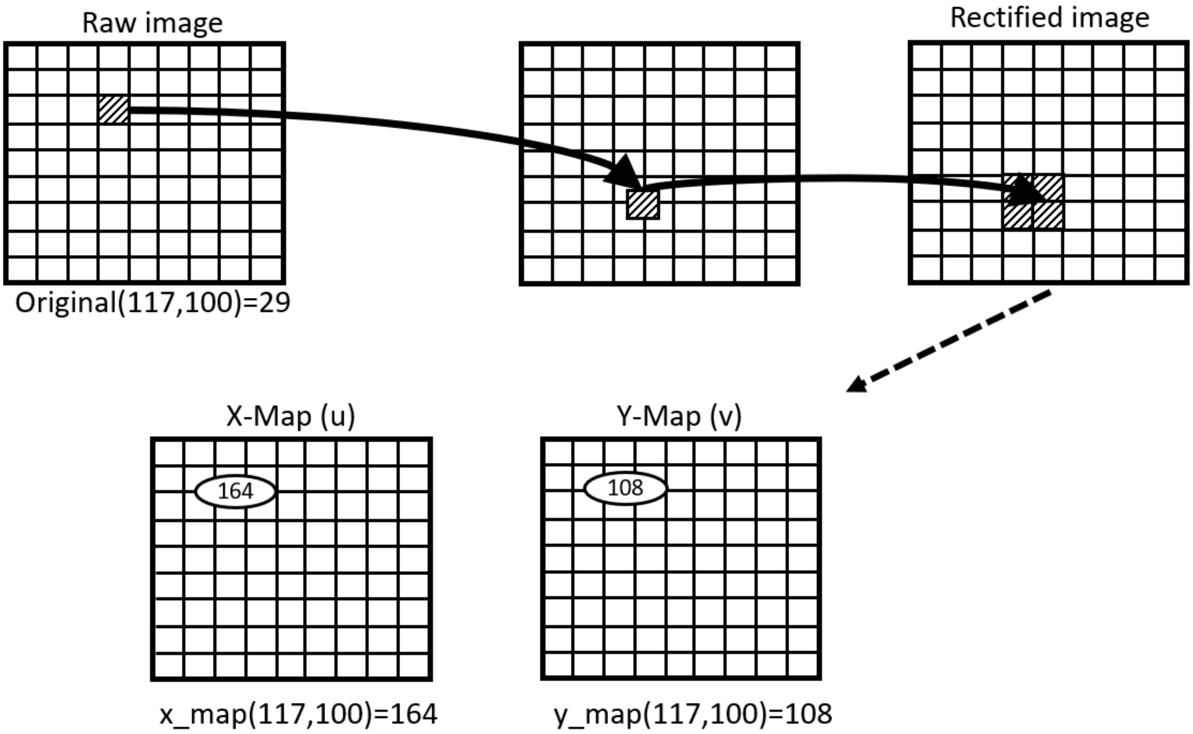


Fig. 4-3: Interpolation Chart

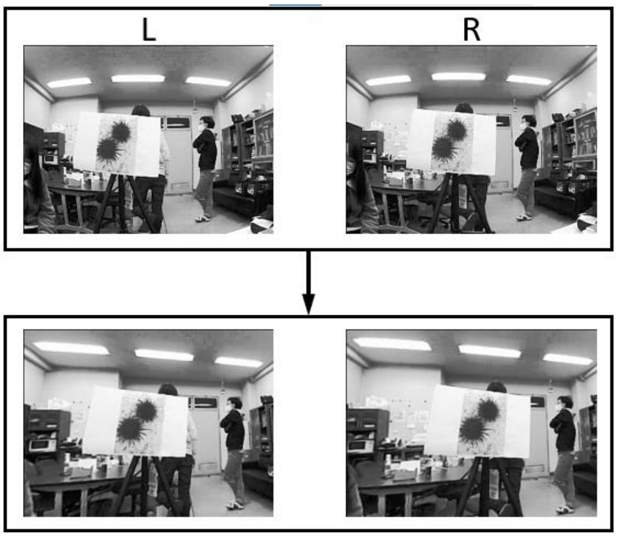


Fig. 4-4: Stereo rectification process diagram.

4.2. Similarity calculation

We get the sea urchin detection frame in the left and right images, and then we need to find the corresponding right detection frame for the left detection frame, which means the corresponding coordinate representation of the same sea urchin in the left and right cameras.

4.2.1. Template Matching

Template matching is the discovery of a small area in the whole image area that matches a given sub-image. The target image is swept from left to right and from top to bottom in turn, and the computer moves the template pixel by pixel from left to right and from top to bottom on the detected image, calculating the match between the template image and the overlapping sub-images. For higher matching degree, the higher the possibility that they are identical.

OpenCV uses squared difference for matching, and the best match is 1. The worse the match, the smaller the match value. In this paper, we use standard squared difference matching[9] with the following Eq. (4-11), where *R* is the output image, *T* is the template, and *I* is the input image.

$$R(x,y)=\frac{\sum_{x',y'}\left(T\left(x',y'\right)-I\left(x+x',y+y'\right)\right)^2}{\sqrt{\sum_{x',y'}T\left(x',y'\right)^2\cdot\sum_{x',y'}I\left(x+x',y+y'\right)^2}} \tag{4-11}$$

In this paper, we use the sea urchin patch 1 recognized in the left image as a template and slide over the sea urchin patch 2 recognized in the right image to find the similarities between the two images. Fig. 4-5 and Fig. 4-6 show the cases with the same patches and different patches of the Template matching, respectively.

(1) Similar patches

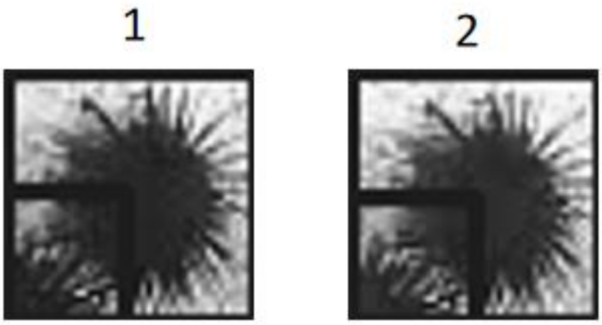


Fig. 4-5: Template matching of same patches.

By comparing the similarity of the same patches, and the detected similarity value is 0.9073.

(2) Different patches

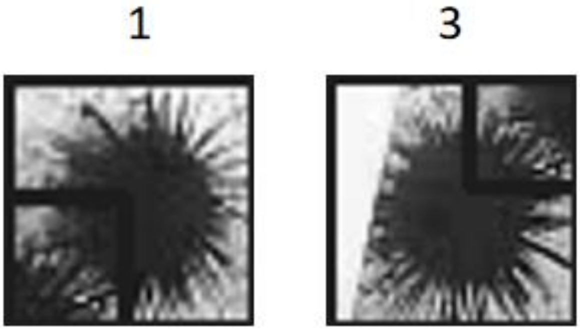


Fig. 4-6: Template matching of different patches.

By comparing the similarity of different patches, and the detected similarity value is 0.5269.

4.2.2. Siamese Network Matching

In this paper, the MNIST database is used to provide the dataset for the Siamese model[13], as shown in Fig. 4-7. On the left side, two example patches are provided for the Siamese model to determine whether these patches belong to the same class. In the middle is the Siamese network model, where the two subnetworks are mirror of each other with the same structure and parameters. If the weights of one subnetwork are updated, the weights of the other subnetwork are also updated. The output of each subnetwork is a fully connected layer. We calculate the Euclidean distance between these outputs and activate them by sigmoid activation so that we can determine the degree of similarity between the two input patches.

We create positive and negative samples from MNIST and then build the Siamese network construct, train the Siamese network on the positive and negative sample pairs using the Siamese network model, and serialize the Siamese network model and the training history graph into a catalog model. Fig. 4-8 and Fig. 4-9 show the similar and different patches of the Siamese network matching, respectively.

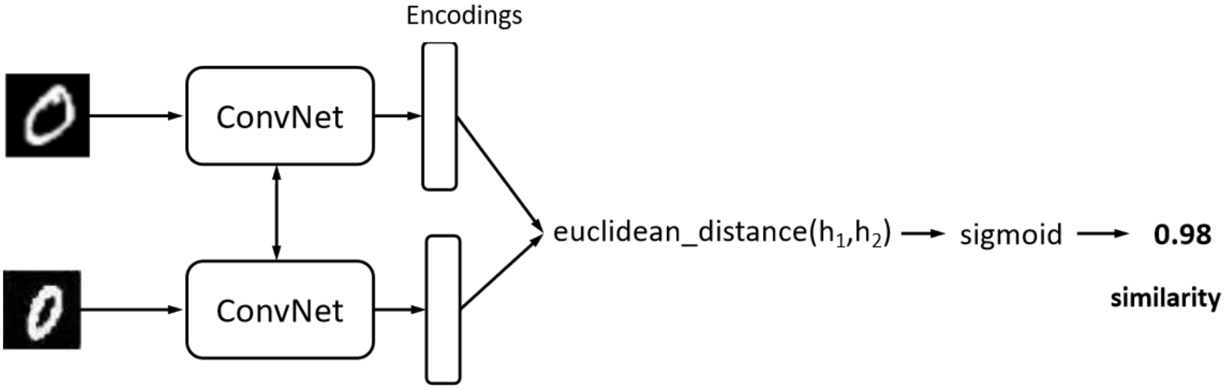


Fig. 4-7: The basic architecture of Siamese network.

(1) Similar patches

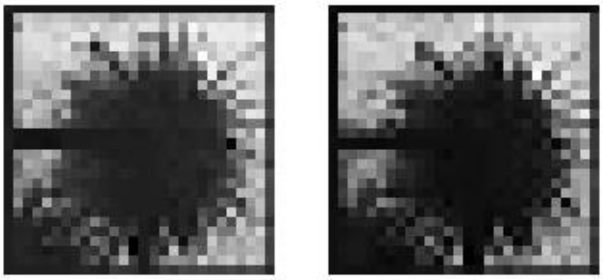


Fig. 4-8: Siamese network matching of same patches.

By comparing the similarity of the same patches, and the detected similarity value is 0.9797.

(2) Different patches

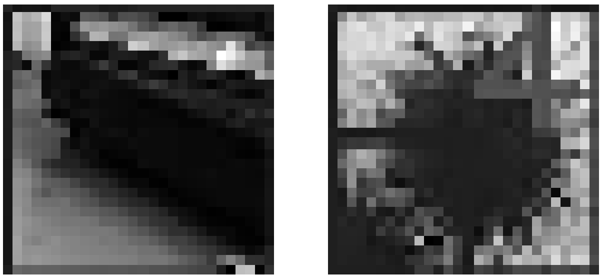


Fig. 4-9: Siamese network matching of different patches.

By comparing the similarity of different patches, and the detected similarity value is 0.000666.

Through experimental comparison, it is found that machine learning takes less running time than deep learning, although the similarity error of deep learning is a little better than machine learning, but combined with time efficiency, this paper uses machine learning to calculate the similarity.

4.3. Stereo Matching

We obtain the pixel coordinate values of the same sea urchin in the left and right cameras respectively, and then we need to perform stereo matching on the left and right images. Matching a three-dimensional point in the left and right camera views, then the disparity value *d* will be calculated on the visual area where the two camera views overlap, so placing the binocular cameras as far forward and parallel as possible will give better results. Since the Hartlry algorithm can only compute the positions of points in the projection transform, the semi-global block matching (SGBM) algorithm is chosen in this paper. SGBM algorithm has four steps: image pre-processing operation, acquisition of generated values, dynamic planning algorithm and image post-processing[14].

(1) Image pre-processing operation

The image pre-processing operation is performed to finally obtain the gradient information of the image, while being able to compensate for the distortion formed by different light intensities. The Sobel operator is used for image preprocessing, and the basic formula is given in Eq. (4-12), where *p(x,y)* is pixel value at *(x,y)*.

$$sobel\left(x,y\right)=2[p(x+1,y)]-p(x-1,y)]+p(x+1,y-1)-p(x-1,y+1)+p(x+1,y+1)-p(x-1,y-1) \tag{4-12}$$

The above equation represents the horizontal Sobel algorithm which subtracts two times the pixel value between neighboring pixels in the same horizontal direction from the lower-right, upper-left, upper-right and lower-left values. Each pixel point in the image will perform Sobel operation, and finally get a new pixel point image P_{NEW}, which is represented by the mapping function in Eq. (4-13).

$$P_{NEW}=\begin{cases} 0; P<-preFilterCap \\ P+preFilterCap;-preFilterCap\le P\le preFilterCap \\ 2\cdot preFilterCap, P\ge preFilterCap \end{cases} \tag{4–13}$$

(2) Matching cost calculation

The gradient cost is obtained by sampling and calculating the gradient information obtained in the previous step, followed by sampling the pre-processed image to obtain the SAD cost, whose cost calculation formula is given in Eq. (4-14), where *L(x+i,y+j)* and *R(x+d+i,y+j)* are the grayscale values of the left image at *(x+i,y+j)* and the right image at *(x+d+i,y+j)*, respectively.

$$C(x,y,d)=\sum_{i=-n}^{n}\sum_{j=-n}^{n}\left|L(x+i,y+j)-R(x+d+i,y+j)\right| \tag{4-14}$$

(3) Dynamic programming

In dynamic programming can effectively suppress this trailing effect, which can both prevent the proliferation of trailing effect and bring some mismatches to the subsequent operations. Therefore, this paper will use SGBM algorithm, which can perform Markov energy transfer by one-dimensional redundant one-dimensional constrained epipolar lines on the image, after which the matching value of a pixel in the image is the superposition of all the neighboring path information around it, and a winner-take-all strategy is used to determine the matching point. The dynamic programming method is shown in Eq. (4-15) and Eq. (4-16).

$$L_r\left(p,d\right) = C\left(p,d\right) + \min(L_r(p-r,d), L_r\left(p-r,d-1\right) + P_1, L_r\left(p-r,d+1\right) + P_1, \min_i L_r(p-r,i) + P_2) - \min_k L_r(p-r,k) \tag{4-15}$$

$$sp(d) = \sum_r L_r\left(p,d\right) \tag{4-16}$$

Where P_1 denotes the probability that the difference between the target pixel point and its surrounding neighboring pixel points is equal to 1, and its concept is expressed by the penalty coefficient. P_2 denotes the probability that the difference between the target pixel point and its surrounding neighboring pixel points is greater than 1, and its magnitude is expressed by the penalty coefficient P_2. P_2 must be greater than P_1, while if the greater the value P_1, P_2 indicates the smoother the image area where its points are located. P_1, P_2 are expressed by Eq. (4-17) and Eq. (4-18) respectively[15].

$$P_1 = 8\times cn\times \mathrm{sg}\,bmSADWindowSize\times sgbmSADWindowSize \tag{4-17}$$

$$P_2 = 32\times cn\times \mathrm{sg}\,bmSADWindowSize\times sgbmSADWindowSize \tag{4-18}$$

Where the parameter *cn* indicates the selected region mapping value in the image and SADWindowSize indicates the SAD window size, here it is generally expressed as an odd number in the range of 3×3 to 21×21. If the size of cn and the parameter SADWindowSize are fixed, the penalty factor P_1 and P_2 indicate two fixed values.

(4) Image post-processing

The image post-processing part is divided into three parts of work: uniqueness detection, left-right consistency check and detecting connected areas.

Uniqueness detection. If the current minimum cost is a multiple of *(1 + uniquenessRatio/100)* of the next lowest cost, the value represented by the store is the disparity value of a point in the requested region, and if this condition is not satisfied, the parallax value of the point is represented by *0*, where uniquenessRatio is a constant. The formula for obtaining the subpixel interpolation is as follows, where d is expressed by Eq. (4-19) and Eq. (4-20) below.

$$denom2 = \frac{\max(sp[d-1] + sp[d+1] - 2 + sp[d], 1)}{16} \tag{4-19}$$

$$d = d + \frac{(sp[d-1] - sp[d+1]) + denom2}{denom2\times 2} \tag{4-20}$$

Left-right consistency check. Taking the left image as the reference image, if the reference image disparity $dispL[x]$ is known, the corresponding right image disparity value $dispR\left[x-d\right]$ can be found, and d represents the difference in horizontal distance between the left and right corresponding images. By finding the minimum matching cost among the data values of $disp\left[x\right]$, the minimum matching surrogate value is its correct disparity value.

Detecting connected areas. A threshold value is set to compare all the pixels with the threshold value. It is determined whether there is a mis-matching point, and there must be at least one point around a detected pixel that meets the connectivity condition. After the detection, the neighboring points around the point are set as the starting points for detection and determination, and the points around these points are found to meet the connectivity conditions, respectively. If the number of detected points exceeds the size of the threshold, the disparity value of the region is set to the correct disparity value, otherwise it is judged as a noisy point and the point is rejected.

In the SGBM matching algorithm, the several parameters are adjusted to obtain the best disparity image. The target point of left image is necessarily generated in the same row in the right image by matching to find the minimum number of disparities, which means the best matching pixel of left image is found in the right image, the effect of SGBM algorithm and the disparity diagram are shown in Fig. 4-10.

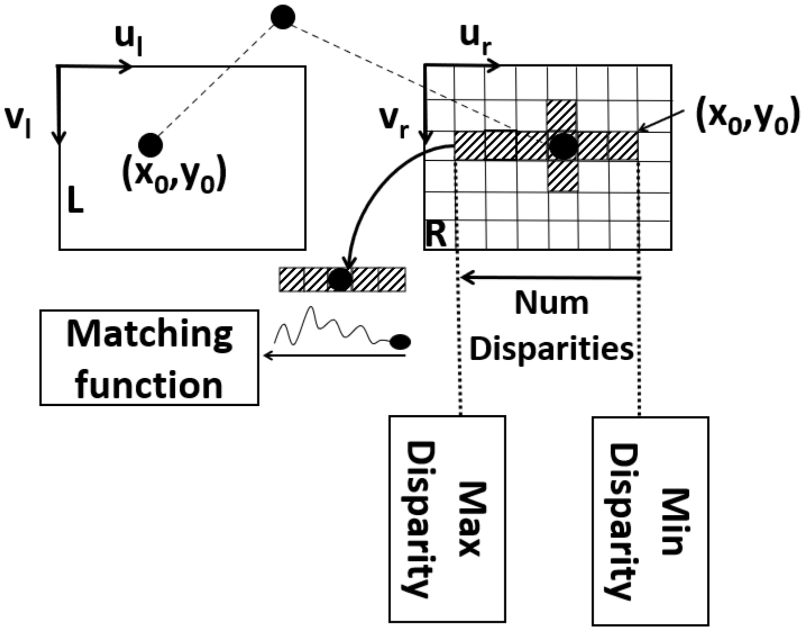


Fig. 4-10: Schematic diagram of the SGBM algorithm.

In OpenCV, depth detection is mainly based on the left image. Assuming that the left image coordinate of the principal point (the coordinate value of the image coordinate origin in the pixel coordinate system) is (u_0,v_0) and the right image coordinate of the principal point is (u'_0,v'_0), and the calculated disparity is d, the three-dimensional depth is obtained from the reprojection matrix Q obtained by stereo correspondence[9] (as shown in Eq. (4-21)), and then the depth can be obtained according to Eq. (4-22) and Eq. (4-23), as shown in Fig.4-11.

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ w \end{bmatrix} = Q \cdot \begin{bmatrix} u \\ v \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -u_0 \\ 0 & 1 & 0 & -v_0 \\ 0 & 0 & 0 & f \\ 0 & 0 & -\frac{1}{T_x} & \frac{u_0 - u'_0}{T_x} \end{bmatrix} \cdot \begin{bmatrix} u \\ v \\ d \\ 1 \end{bmatrix} \tag{4-21}$$

$$d = X_L - X_R \tag{4-22}$$

$$depth = \frac{z_w}{w} = -\frac{f \cdot Tx}{d} \tag{4-23}$$

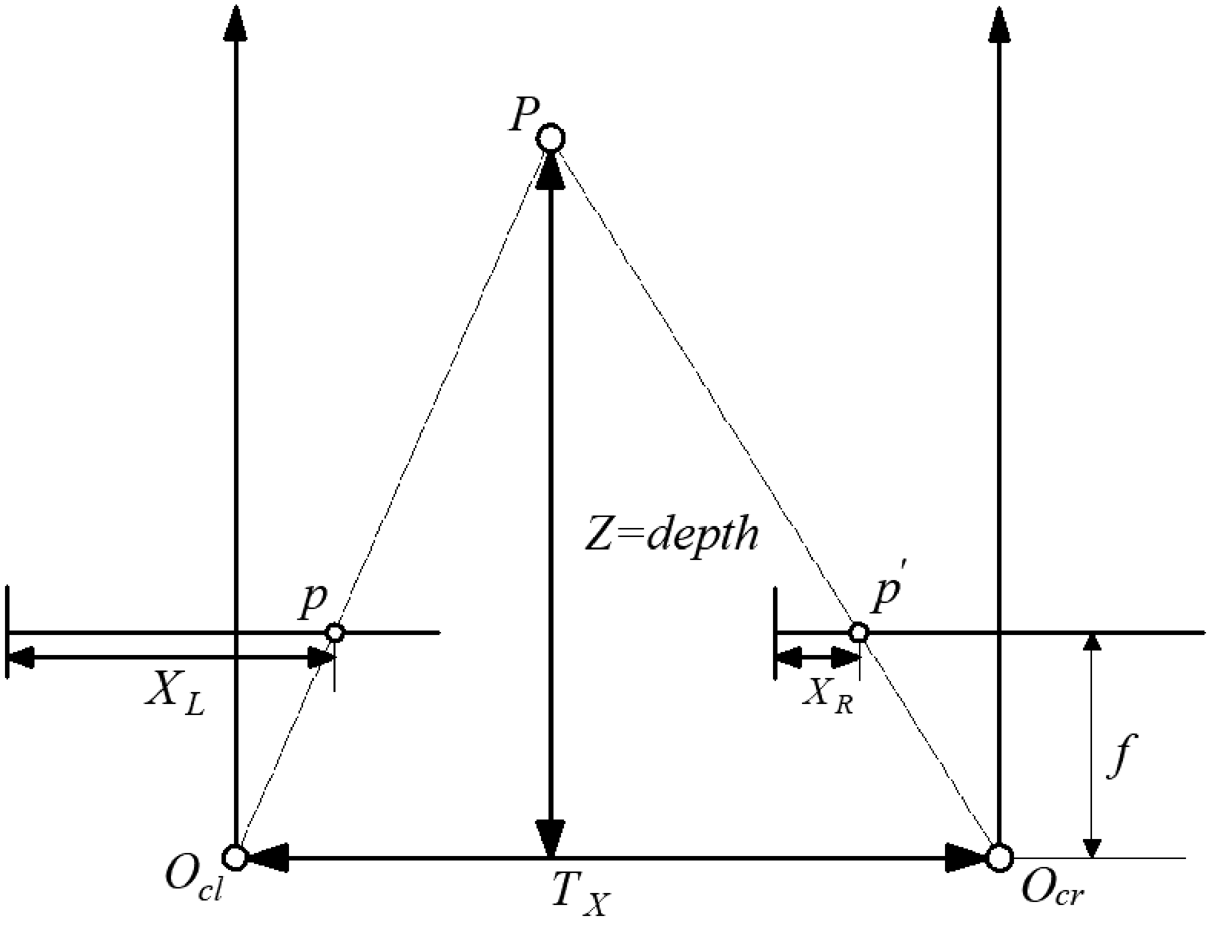


Fig. 4-11: Geometric schematic of binocular ranging.

5. Experimental results and analysis

Firstly, this paper uses the specified distance depth detection of sea urchin pictures in air (laboratory) using a binocular stereo vision ranging system, mainly to adjust each parameter by air experiments in order to obtain the optimal parameters. Finally, the best parameters obtained are used to detect the depth distance of real sea urchins underwater, and the results are analyzed.

5.1. Air experimental

In this paper, three sets of better calibrated parameters obtained in Chapter 2 are used for depth detection. Firstly, experiments were conducted on the mounted system in an air environment. The sea urchin picture was tested at a certain distance using a tripod fixed as shown in the Fig. 5-1. The tripod was fixed at a distance from 500 to 1200 mm, respectively, and depth experiments were conducted at 100 mm intervals.

We obtain the depth information of the detected sea urchin by using the pixel coordinate values of the corresponding detection frame taken by the binocular camera, which is the direct distance between the sea urchin and the cameras. For the convenience of calculation, the actual measured distances were integers. The depth error equation is shown in Eq. (5-1), where d is the detected depth and *l* is the real distance.

$$Depth\_error = \frac{|d-l|}{l} \times 100\% \qquad (5\text{-}1)$$

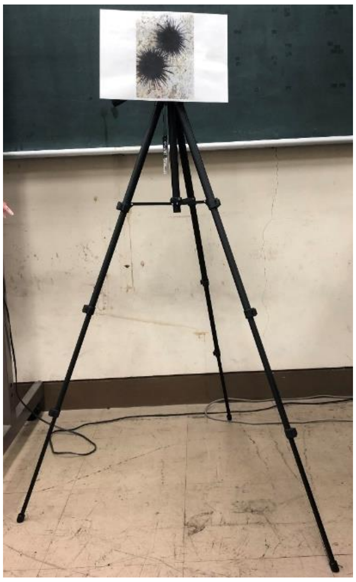


Fig. 5-1: Sea urchin picture tripod.

5.1.1. Select optimal calibration parameters

Three better sets of calibration parameters were obtained in Chapter 2, and we used them to perform depth detection for sea urchin photos at different distances[16]. Since the machine learning can detect many suspected sea urchin detection frames within a specified distance at same time, a large amount of datum can be obtained. To make it easier to judge the accuracy of the data and use all the detected data information, this paper uses the extraction of the median (middle number) of all data to observe the accuracy error of the system. All calibration results are shown in Fig. 5-2.

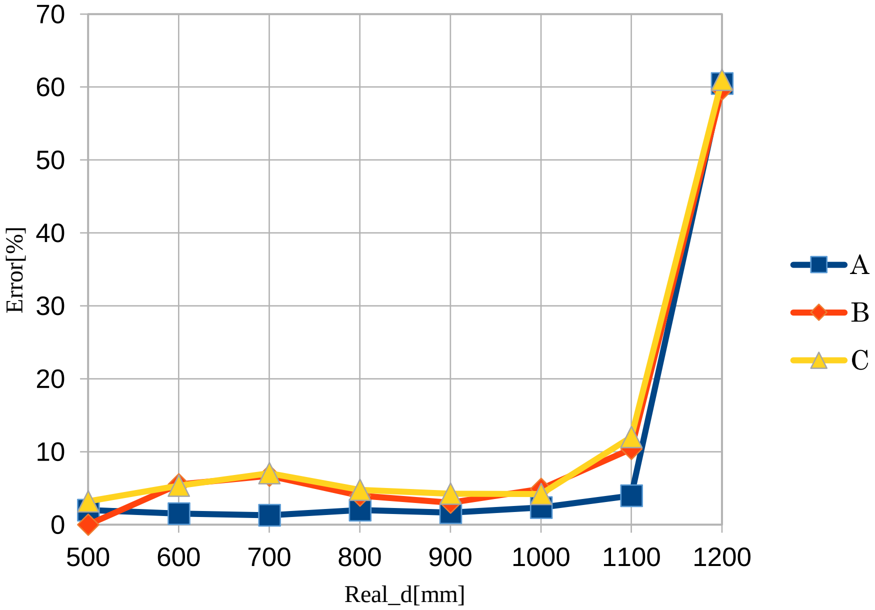


Fig. 5-2: All calibration results error graph.

Because there are many interferences in the laboratory environment, there will be objects that are not sea urchins misidentified by the classifier, and pictures of sea urchins are placed in the middle of the camera lens, so in this paper, 100~200 pixel horizontal coordinates in the detection frame are selected as the correct sea urchins (the camera frame size is 320×240), and the detection results after selection are shown in Fig. 5-3, and subsequent studies will only show the result graphs in this case.

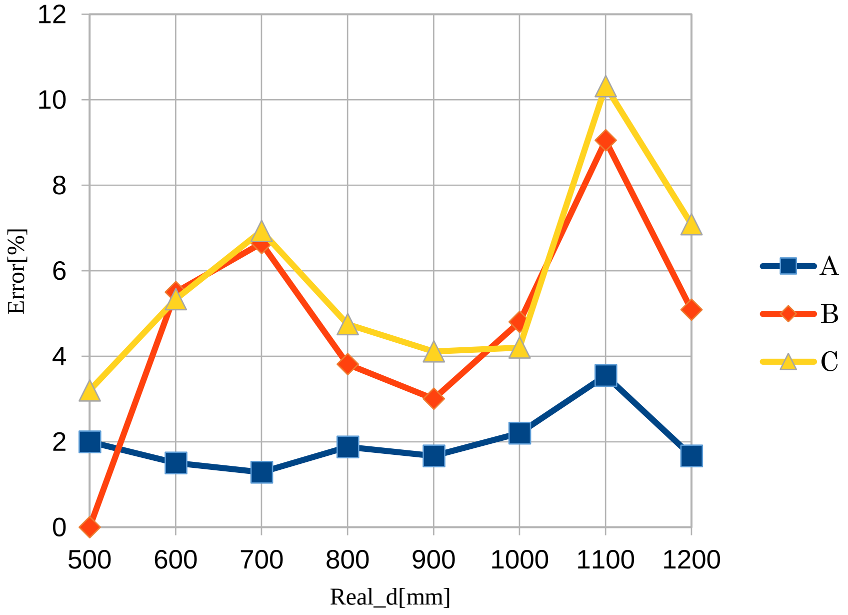


Fig. 5-3: Selected calibration result error graph.

Fig. 5-4 shows the graphs of the time-consuming results for the three sets of calibration parameters at different distances. Since the duration of the videos detected at different distances is different, there is a contrast between the three calibration results at the same video.

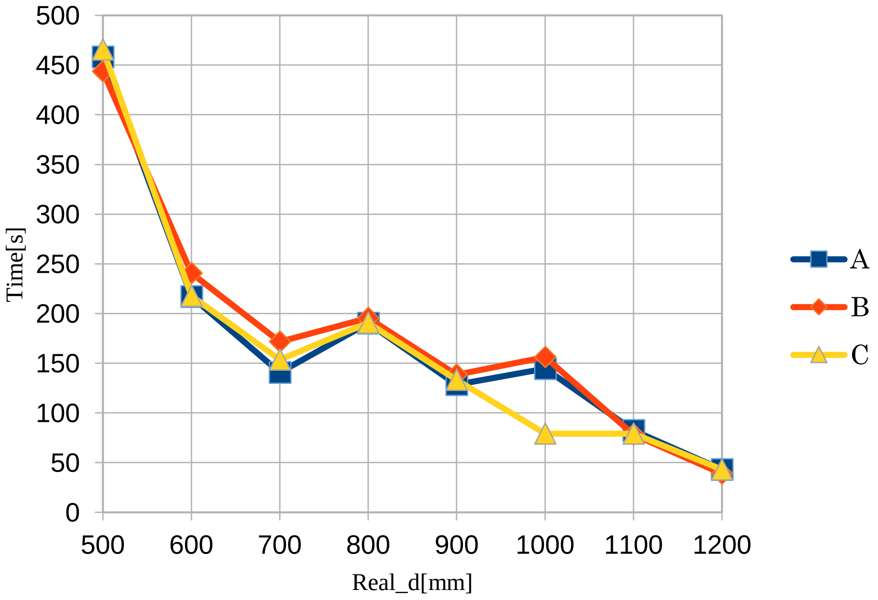


Fig. 5-4: Different calibration parameters time consumption graphs.

According to the experimental results, we can see that the three sets of calibration parameters have achieved good accuracy, and the accuracy error is less than 12%, and the best set of accuracy error can reach less than 5%. Fig. 5-5 shows the disparity maps produced by the three sets of calibration parameters for the same pair of pictures.

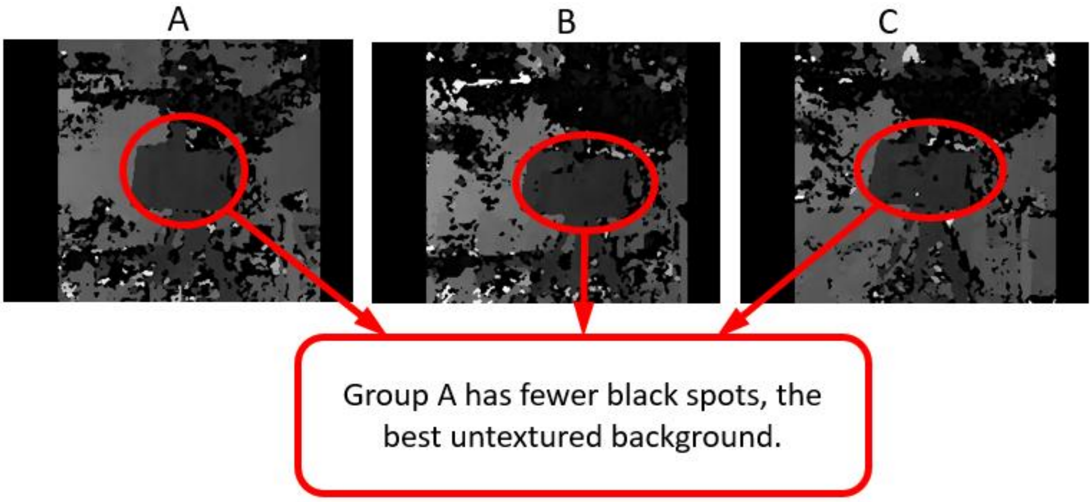

A
B
C
Group A has fewer black spots, the best untextured background.

Fig. 5-5: Comparison of the disparity maps of the three group calibration parameters (black is invalid).

Through comparison, it can be found that group A produces the best disparity map cloud continuity and the best error accuracy among the three calibration parameters, so group A is chosen as the optimal calibration parameter in this paper.

5.1.2. Image enhancement processing

Base on the result above, so we use the group A parameters for subsequent research. According to 3.1.2, we use the histogram equalization to enhance the image detail. Fig. 5-6 below shows the comparison between the two cases of enhanced processing(hist) and unprocessed, and Fig. 5-7 shows the time consuming of both.

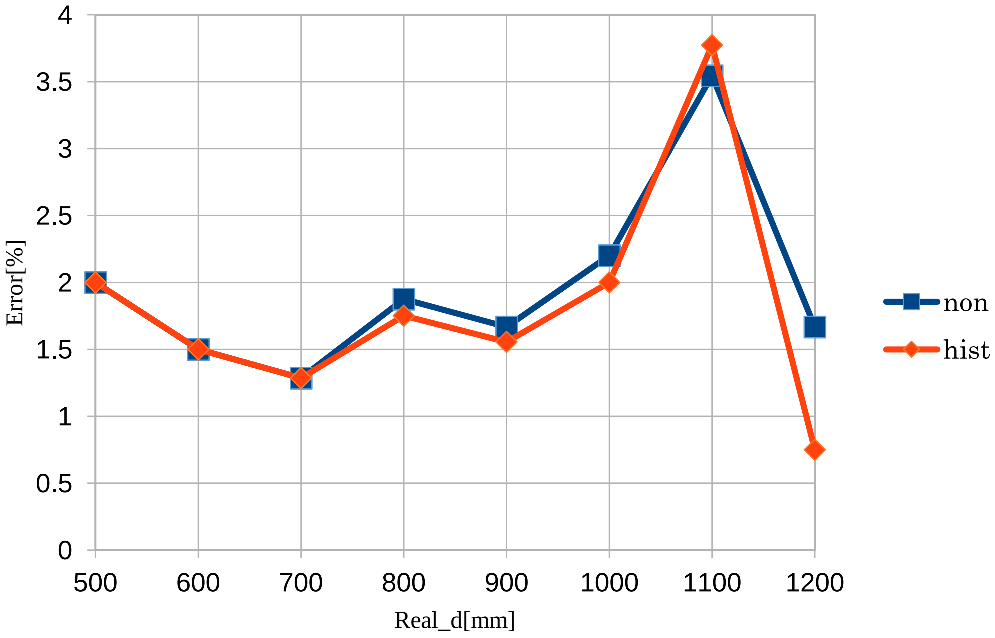


Fig. 5-6: Image enhancement compare graph.

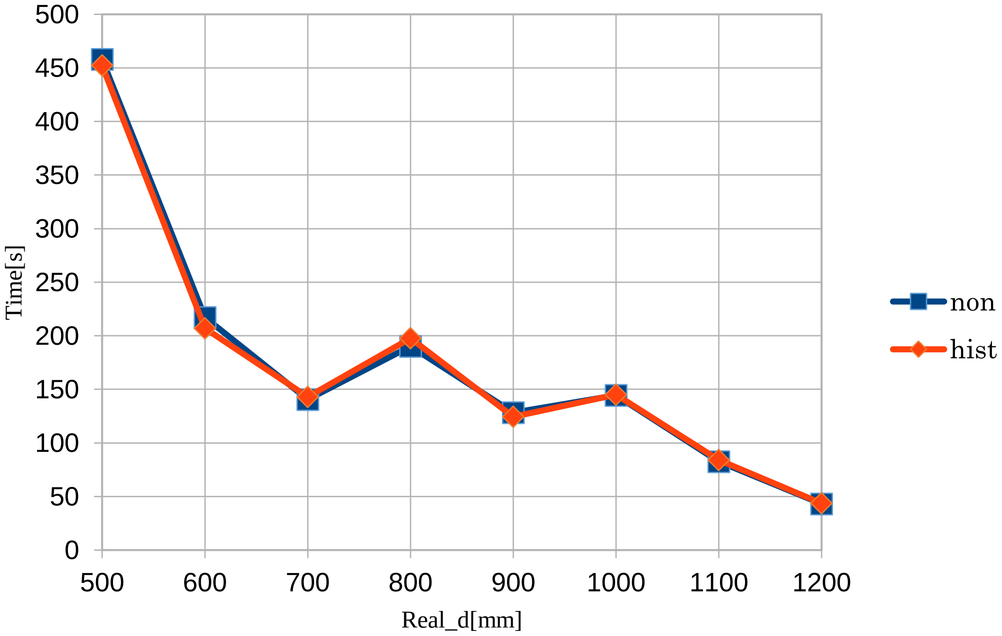


Fig. 5-7: Image enhancement compare graph.

From the above result graphs, we can see that although the error of the enhanced image is larger at 1100 mm than that of the unprocessed one, but the error is smaller at all other distances than that of the unprocessed one. Considering this, we choose to add image enhancement in the image pre-processing.

5.1.2. Select filter type

According to 3.1.3, we need to filter the images captured by the camera. First, we need to determine the position of the filter, here we divide into three stages: before image rectification, after rectification and after disparity map generation. In order to control the variable singularity, here the same set the number of filter cores as 3.

(1) Before image rectification

Table 5-1 Depth errors of different filter types at different distances before image rectification.

Real_d(mm)	non	blur	median	gaussian	bilateral
500	2	2	2	2	2
600	1.5	1.5	1.66667	1.5	1.5
700	1.28571	1.28571	1.42857	1.42857	1.28571
800	1.875	1.875	1.75	1.875	1.75
900	1.66667	1.77778	1.66667	1.77778	1.55556
1000	2.2	2.4	3.1	2.5	2.3
1100	3.54545	3.63636	4.13636	3.63636	3.54545
1200	1.66667	2.25	2	2.91667	1.83333

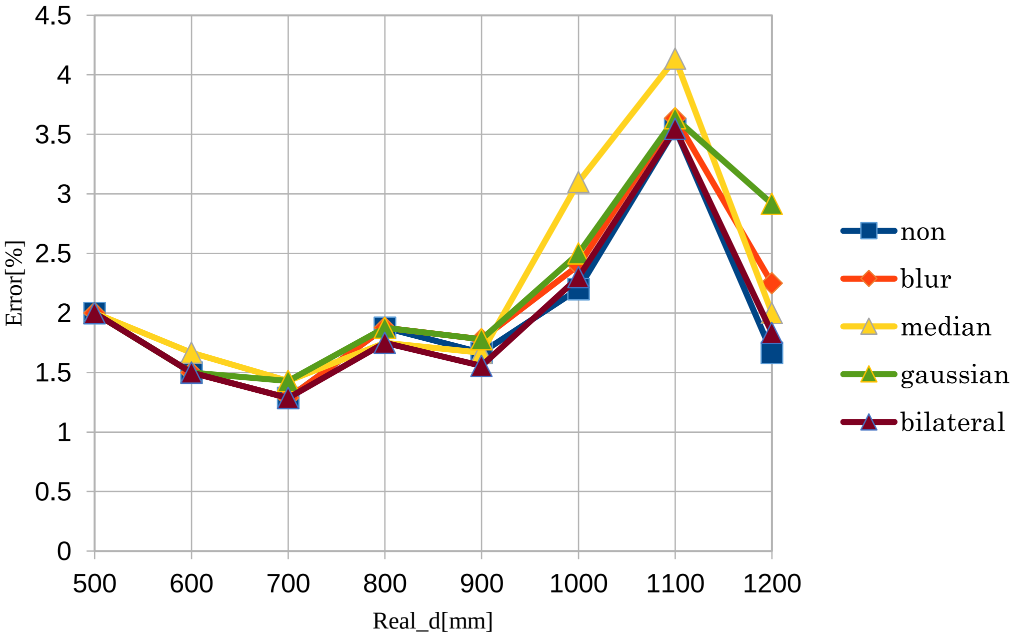


Fig. 5-8: Error comparison graph for different filter types before image rectification.

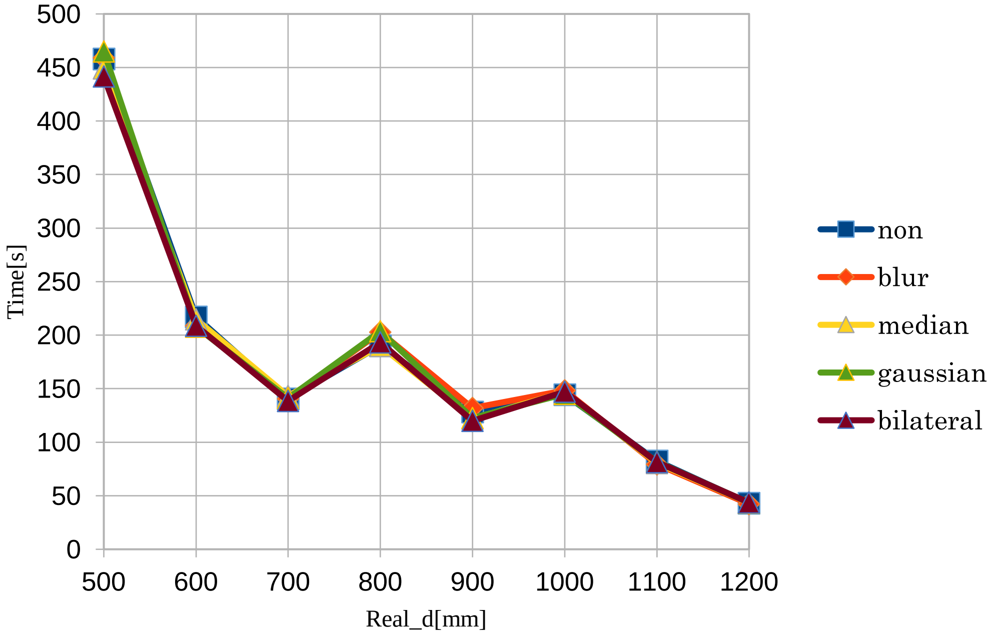


Fig. 5-9: Time consuming graphs for different filter types before image rectification.

Based on the above result data, we can find that in terms of error accuracy, the filter is currently more stable than the unprocessed image only for the bilateral filter, but the change is not very significant. In terms of time consuming, the increase of the filter is beneficial to reduce the running time of the code and to improve the efficiency.

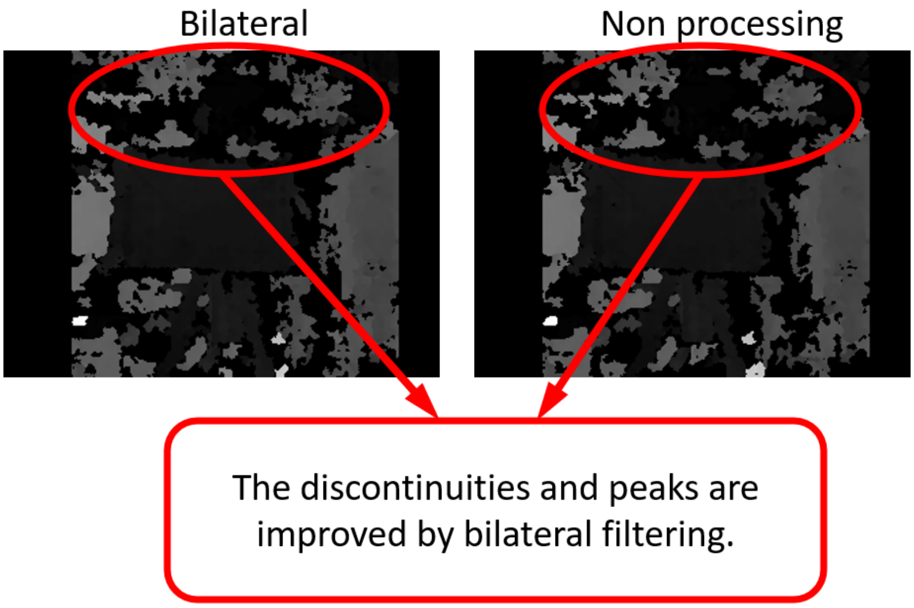

Bilateral
Non processing
The discontinuities and peaks are improved by bilateral filtering.

Fig. 5-10: Comparison of the disparity maps of bilateral and non-processing before image rectification.

Fig. 5-10 shows the disparity map produced after bilateral filtering before image rectification compared with the unprocessed disparity map. By comparison, we can find that the texture of the disparity map generated after the bilateral filtering is better than unprocessed one, but it is not obvious. This also corresponds to the comparison data in Table 5-1.

(2) After rectification

Table 5-2 Depth errors of different filter types at different distances after rectification.

Real_d(mm)	non	blur	median	gaussian	bilateral
500	2	2	2	2	2
600	1.5	1.5	1.66667	1.5	1.5
700	1.28571	1.42857	1.42857	1.28571	1.42857
800	1.875	1.875	1.625	1.875	1.875
900	1.66667	1.77778	1.55556	1.66667	1.55556
1000	2.2	2.5	2.9	2.4	2.2
1100	3.54545	3.59091	3.77273	3.54545	3.72727
1200	1.66667	1.5	2.08333	1.66667	1.66667

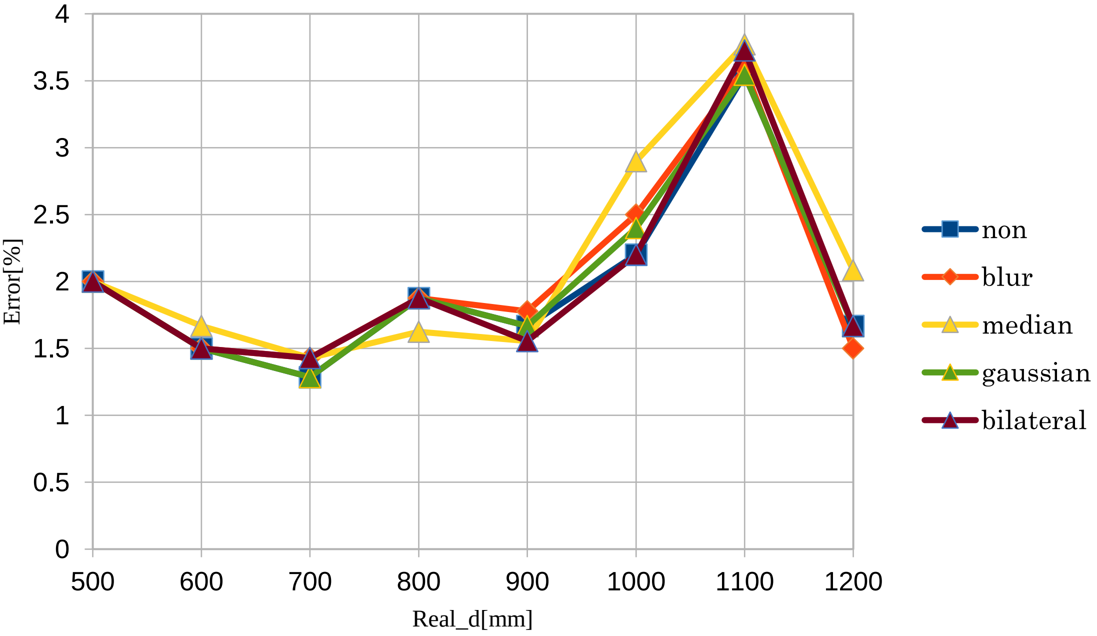


Fig. 5-11: Error comparison graph for different filter types after rectification.

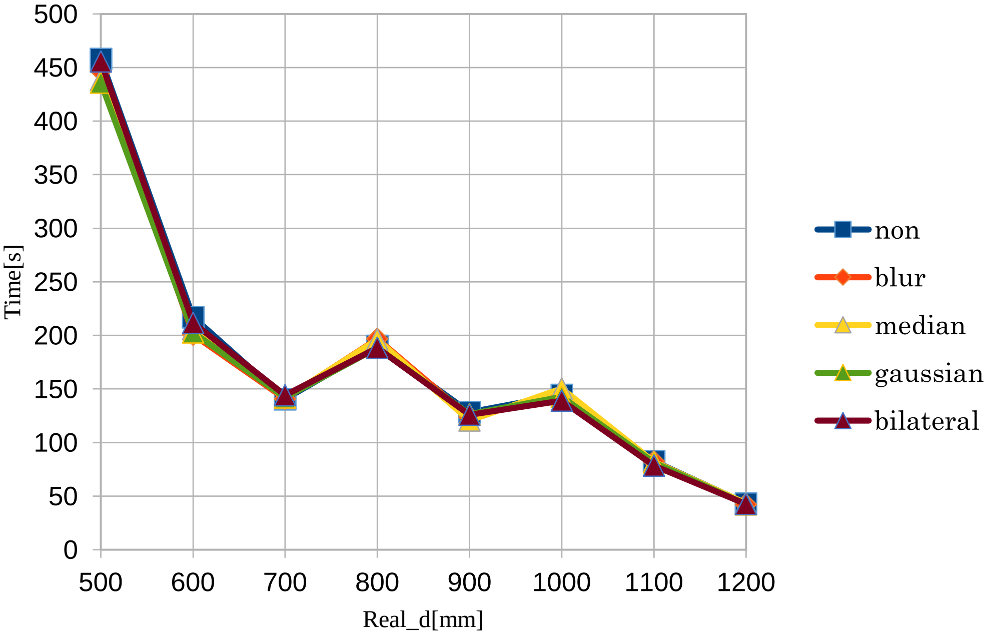


Fig. 5-12: Time consuming graphs for different filter types after rectification.

Based on the above result data (especially Table 5-2), we can see that filtering the image after image rectification does not have much effect on the depth error, so this paper does not take any processing for the rectified image.

(3) After disparity map generation

Table 5-3 Depth errors of different filter types at different distances after disparity map generation.

Real_d(mm)	non	blur	median	gaussian	bilateral
500	2	2	2	2	2
600	1.5	1.33333	1.5	1.5	1.5
700	1.28571	1.28571	1.28571	1.28571	1.28571
800	1.875	1.625	1.75	1.75	1.75
900	1.66667	1.55556	1.66667	1.55556	1.66667
1000	2.2	2.2	2	2.2	2.2
1100	3.54545	3.31818	3.5	3.40909	3.5
1200	1.66667	1.66667	1.75	1.66667	1.66667

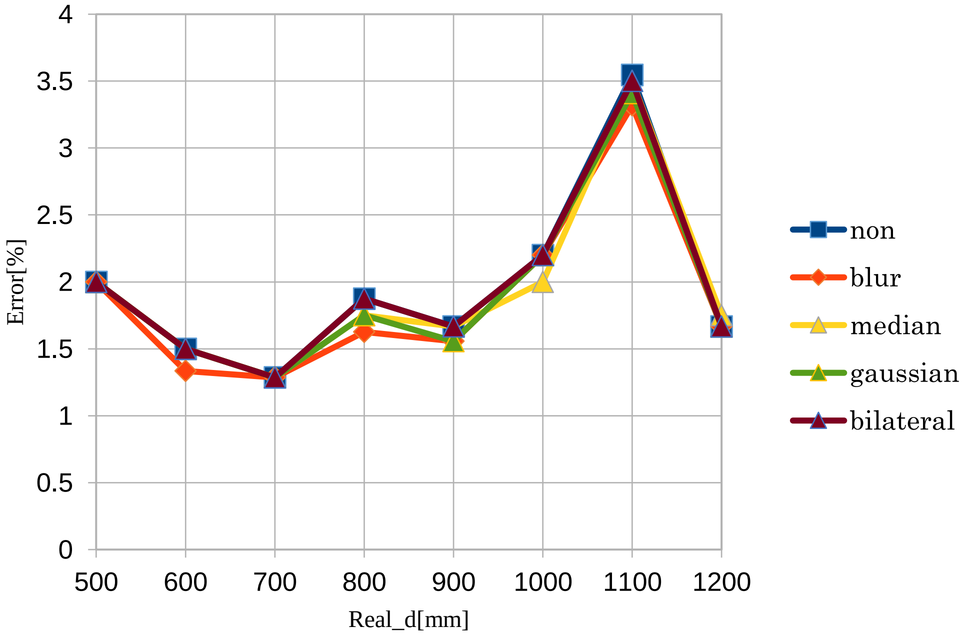


Fig. 5-13: Error comparison graph for different filter types after disparity map generation.

Table 5-4 Time consuming of different filter types at different distances after disparity map generation.

Real_d(mm)	non	blur	median	gaussian	bilateral
500	458.033	447.415	455.634	454.1621	457.497
600	217.122	206.962	213.364	210.865	213.772
700	140.587	140.177	141.539	142.542	144.077
800	190.278	191.068	194.595	196.0677	190.022
900	128.458	129.438	127.793	126.7254	126.312
1000	144.456	146.892	146.639	140.2694	152.57
1100	82.4577	81.7454	82.4628	85.87254	84.7418
1200	43.1064	44.0302	44.4993	44.68589	44.3215

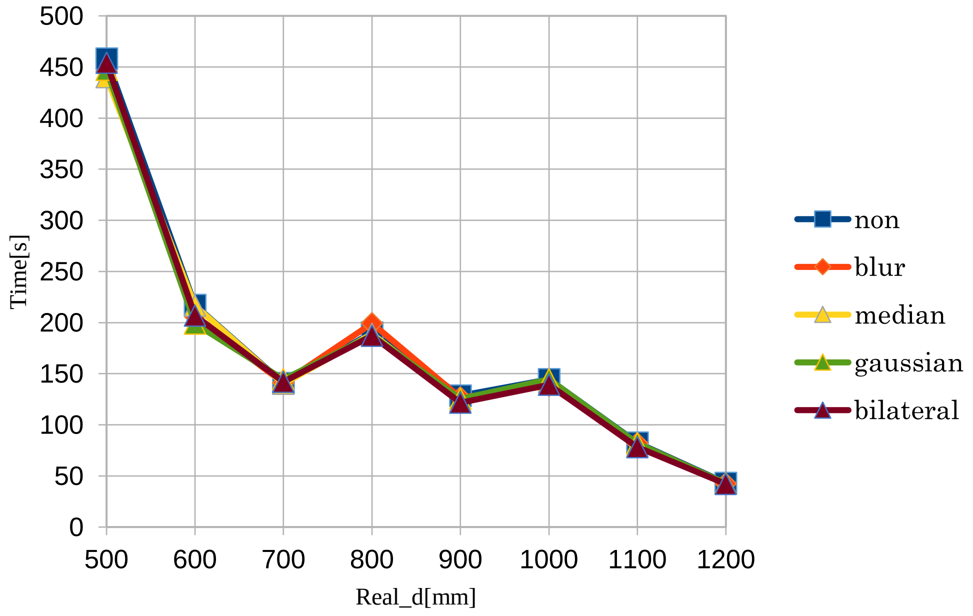
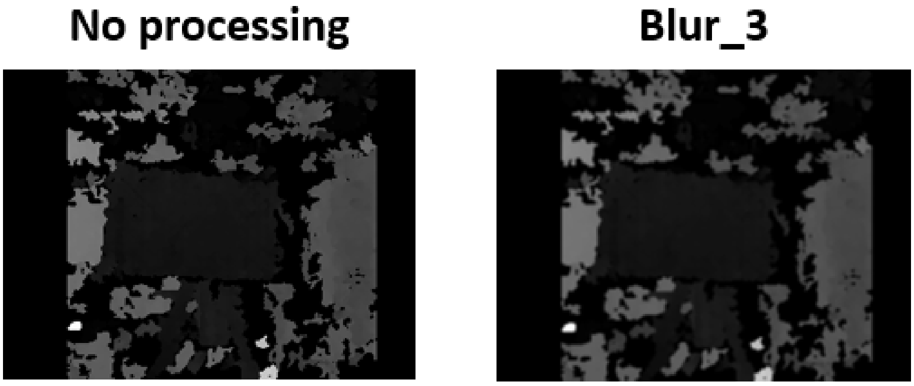


Fig. 5-14: Time consuming graphs for different filter types after disparity map generation.

According to the above result data can be seen, we can find that the depth error after blur filtering has been significantly improved, and the time consumption is also relatively good, so this paper chooses to use blur filter at this stage.



No processing

Blur_3

The edges of the disparity map are clearly blurred.

Fig. 5-15: Comparison of the disparity maps of blur and non-processing after disparity map generation.

According to Fig. 5-15, we can see that the edges of the disparity map are blurred, so it shortens the disparity values of adjacent pixels, which helps prevent sudden abrupt changes in disparity values and reduces noise.

5.1.3. Select the optimal filtering parameters

Based on the conclusions in 5.1.2, we already know what type of filter to set at which stage to achieve the best results. In this subsection, we continue with the selection of the best filter parameters from the obtained result data.

(1) Before image rectification

Before image rectification, we need to apply bilateral filtering to the disparity map. In this paper, we will discuss the parameter range from 3 to 8, where 0 shows the case without the bilateral filtering process.

Table 5-5 Depth errors of bilateral filters in different parameters before image rectification.

Real_d(mm)	0	3	4	5	6	7	8
500	2	2	2	2	2	2	2
600	1.5	1.5	1.5	1.5	1.5	1.5	1.5
700	1.28571	1.28571	1.28571	1.28571	1.14286	1.14286	1.28571
800	1.875	1.75	1.75	1.75	1.75	1.75	1.75
900	1.66667	1.55556	1.66667	1.66667	1.66667	1.66667	1.55556
1000	2.2	2.3	2	2	2.3	2.3	2
1100	3.54545	3.54545	3.81818	3.81818	3.68182	3.68182	3.68182
1200	1.66667	1.83333	1.75	1.75	1.5	1.5	1.5

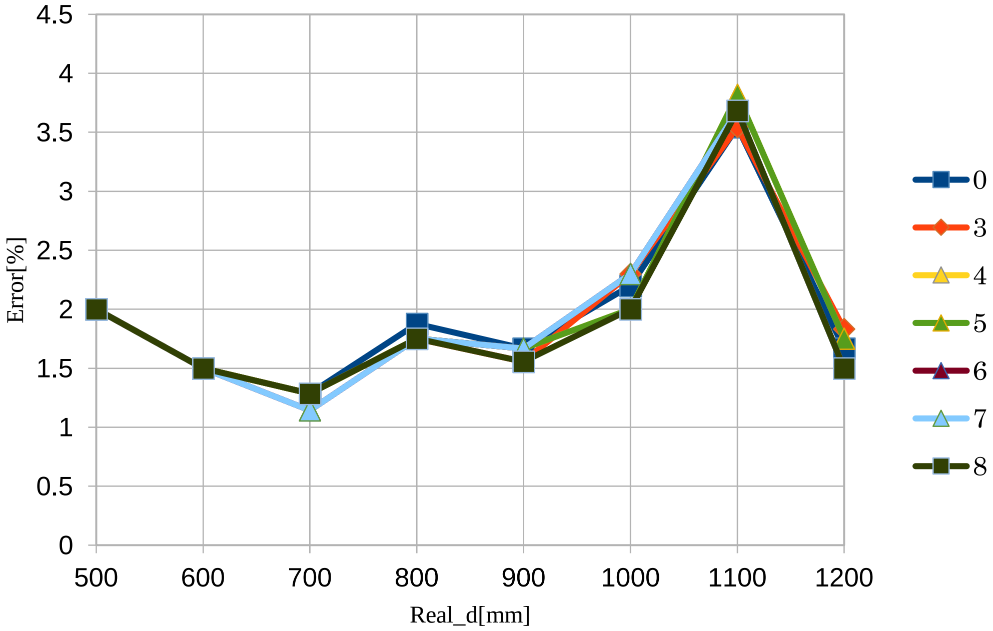


Fig. 5-16: Error graph for bilateral filters at different parameters before image rectification.

Table 5-6 Time consuming of bilateral filtering on different parameters before image rectification.

Real_d(mm)	0	3	4	5	6	7	8
500	458.033	441.477	436.971	476.614	494.796	462.973	471.799
600	217.122	208.829	200.808	217.202	228.167	227.03	220.973
700	140.587	138.528	140.022	149.245	155.642	152.536	148.231
800	190.278	193.233	193.546	207.455	209.965	206.471	206.018
900	128.458	120.058	130.028	127.825	133.055	134.862	130.487
1000	144.456	147.267	154.605	160.052	165.106	151.168	160.382
1100	82.4577	81.2675	87.5649	88.1262	87.3719	86.5646	85.3889
1200	43.1064	43.6327	47.6569	46.5721	45.9128	45.8117	47.7525

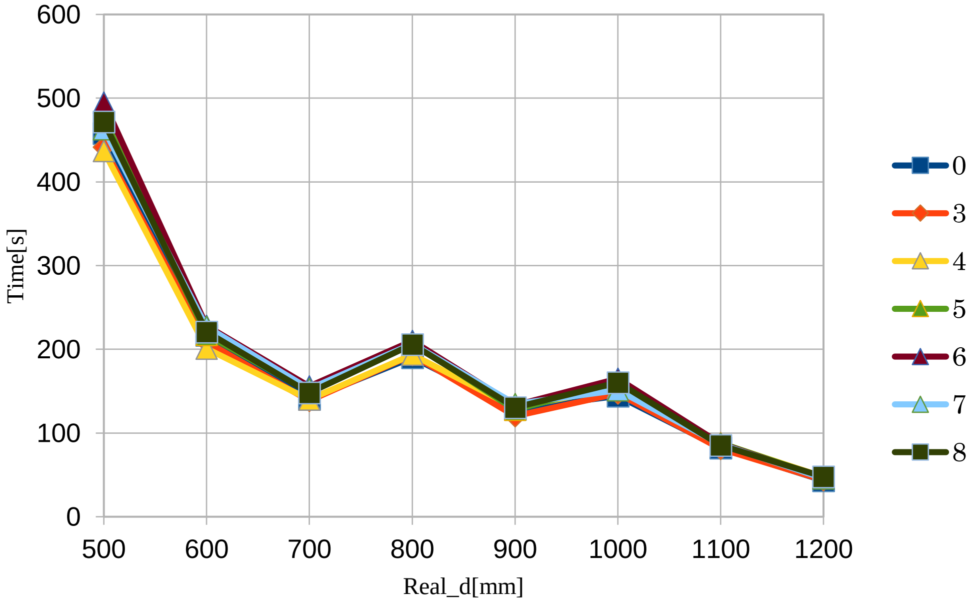
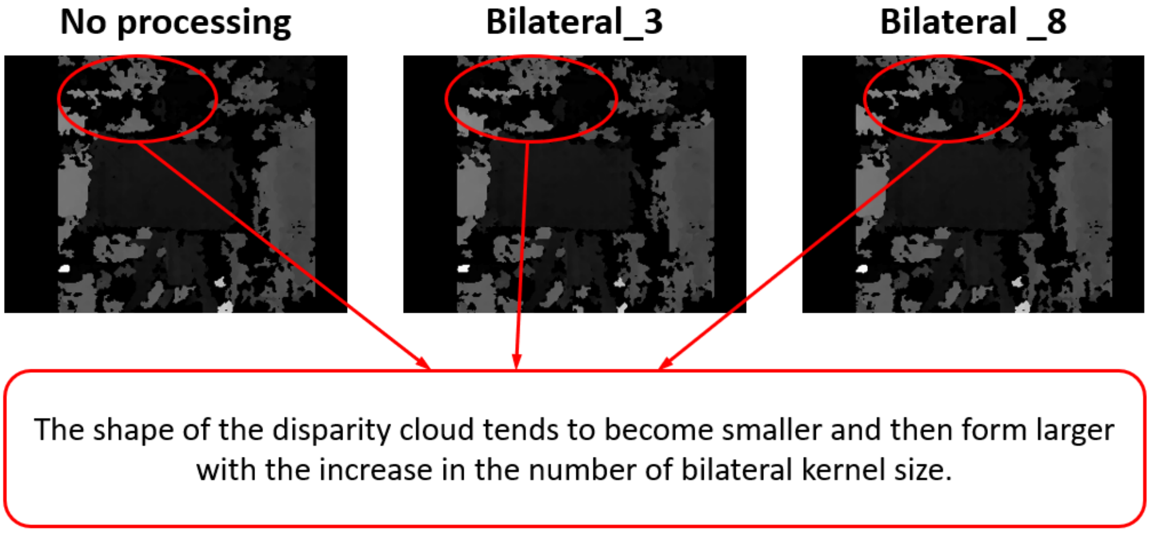


Fig. 5-17: Time consuming graph of bilateral filtering at different parameters before image rectification.

Based on these result data, we can find that the depth error of parameter 3 is more stable and optimal. For the time consuming, parameter 3 also takes a more desirable result. Therefore, the bilateral filtering parameter in this paper is chosen as 3.



The shape of the disparity cloud tends to become smaller and then form larger with the increase in the number of bilateral kernel size.

Fig. 5-18: Comparison of disparity map with different parameters of bilateral filtering.

Through Fig. 5-18, we can find some small changes in the untextured background of the disparity map, so changing the parameter variation can effectively improve the effect of the disparity map.

(2) After generating the disparity map

After generating the disparity map, we need to blur filter the images. In this paper, we will compare the results of parameters 2~11 respectively and select the optimal parameters by comparing the result data.

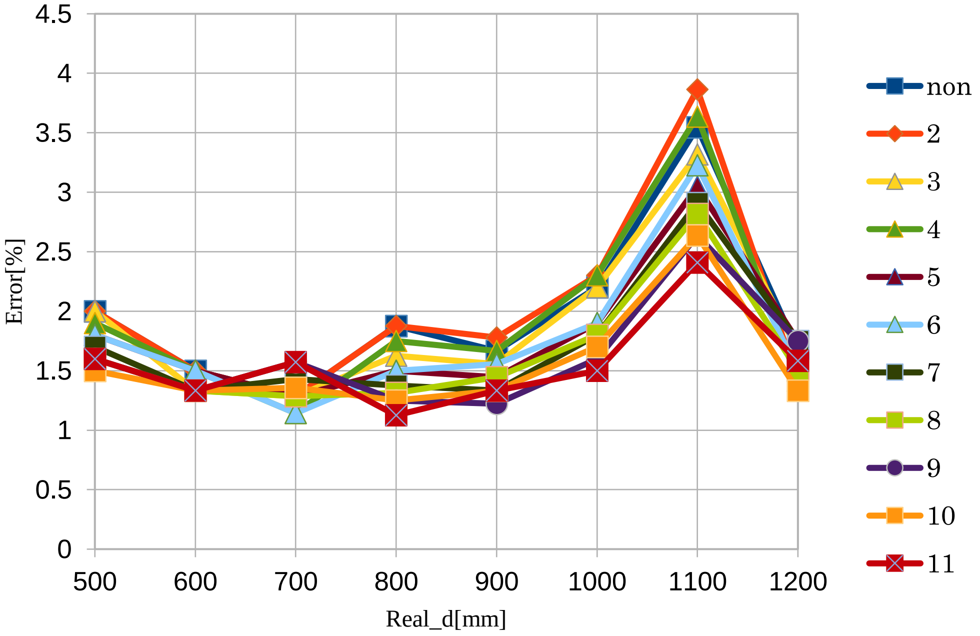


Fig. 5-19: Error graph for blur filters at different parameters after generating the disparity map.

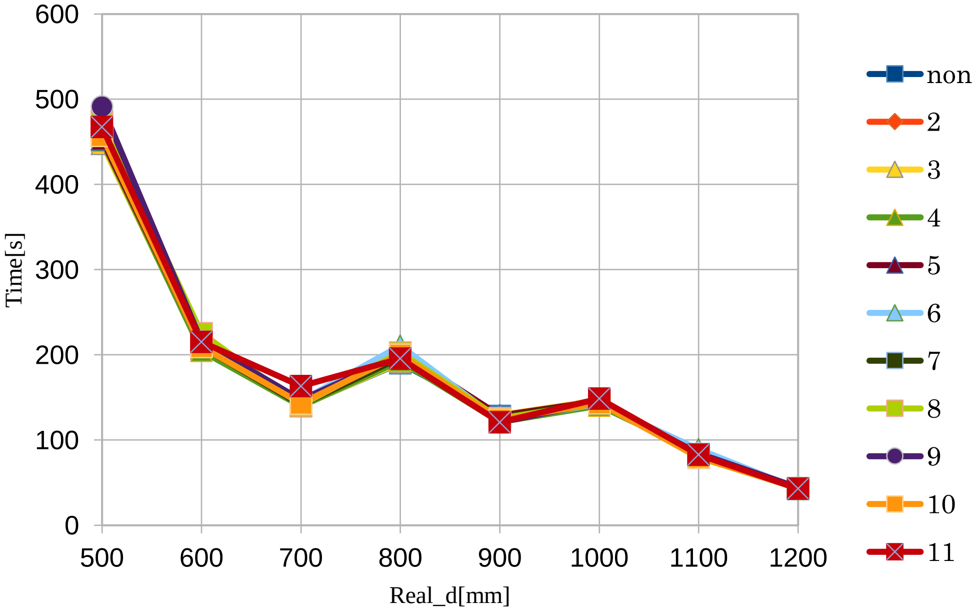


Fig. 5-20: Time consuming graphs for blur filters at different parameters after generating the disparity map.

From the result figures, we can see the most stable trend of depth error for parameter 10. For depth error and time-consuming results, we find that parameter 10 enables the result data to be optimal.

5.1.5 Optimal image pre-processing

According to our previous research, we use histogram equalization, Bilateral filter, and Blur filter to process images, which are mainly used to enhance image information and remove image noise.

Table 5-7 Comparison of depth error results after non-processing and image processing at different distances.

Real_d(mm)	non	hist	best_parameter	best+hist
500	2	2	1.6	1.6
600	1.5	1.5	1.16667	1.16667
700	1.28571	1.28571	1.42857	1.42857
800	1.875	1.75	1.25	1.25
900	1.66667	1.55556	1.33333	1.33333
1000	2.2	2	1.7	1.4
1100	3.54545	3.77273	2.90909	2.95455
1200	1.66667	0.75	1.08333	1

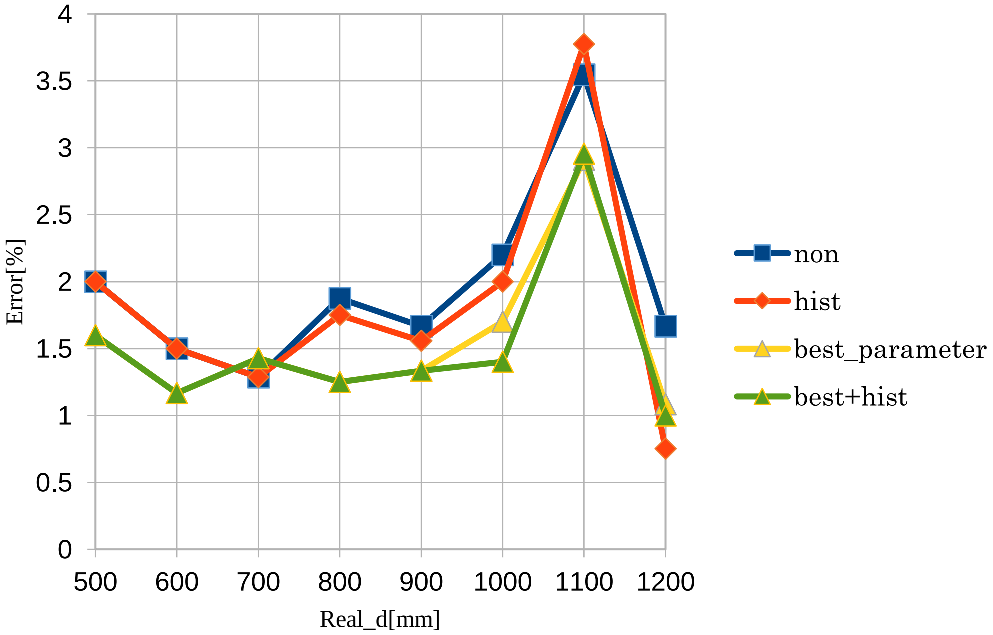


Fig. 5-21: Comparison graph of depth error results after non-processing and image processing.

Table 5-8 Comparison of the time consuming after non-processing and image processing at different distances.

Real_d(mm)	non	hist	best_parameter	best+hist
500	458.033	452.208	455.761	486.734
600	217.122	207.019	201.056	220.091
700	140.587	142.933	142.852	151.126
800	190.278	197.68	193.89	203.659
900	128.458	124.572	123.051	127.505
1000	144.456	145.184	142.298	152.552
1100	82.4577	84.1546	81.0036	87.1071
1200	43.1064	43.6221	43.1393	43.7924

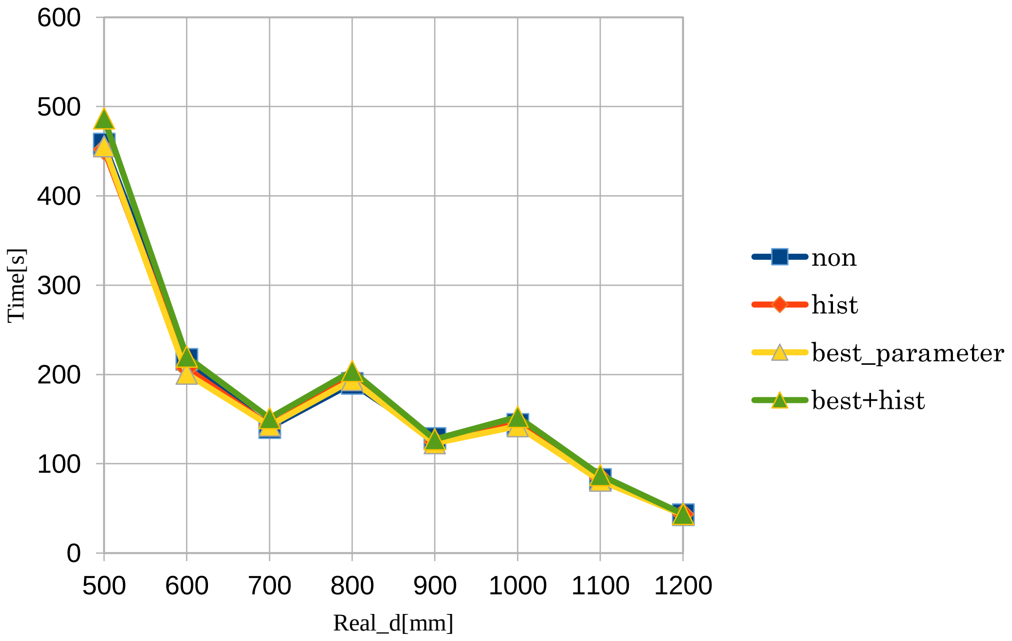


Fig. 5-22: Comparison graph of non-processing and image processing time consuming.

From the above result graphs, this paper obtains better parameters data, and the depth error can be effectively reduced by the image processing. The result data shows that hist can effectively reduce the depth error value, but hist will increase the time consuming; In addition to the filter can effectively reduce the depth error value, it will also reduce the time consuming, but the parameters of the filter need to find the optimal parameter value by comparison.

5.1.4. Select optimal stereo parameters

After the above image processing, we also need to choose the optimal stereo parameters. The current SGBM function parameters of the OpenCV library can be modified by the following four aspects: (1) Mindisparity; (2) BlockSize; (3) NumDisparity; (4) Mode. Since the time consuming does not differ significantly in terms of parameter variation, the next parametric analysis does not analyze the time consuming specifically.

(1) Mindisparity

Table 5-9 Depth errors for different Mindisparity.

Real_d(mm)	28	29	30	31	32	33	34	35	36	37
500	5	1.4	1.6	1.8	1.8	2	2	2	1.2	1.2
600	1.1667	1.1667	1.1667	1.1667	1.1667	1.1667	1.1667	1.1667	1.3333	1.3333
700	1.4286	1.4286	1.4286	1.4286	1.4286	1.4286	1.4286	1.4286	2	2
800	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.3125	1.375	1.375
900	1.3333	1.3333	1.3333	1.3333	1.3333	1.3333	1.3333	1.3333	1.7222	1.7222
1000	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.3	1.3
1100	2.9546	2.9546	2.9546	2.9546	2.9546	2.9546	2.9546	2.9546	2.6364	2.6364
1200	1	1	1	1	1	1	1	1	1.5833	1.75

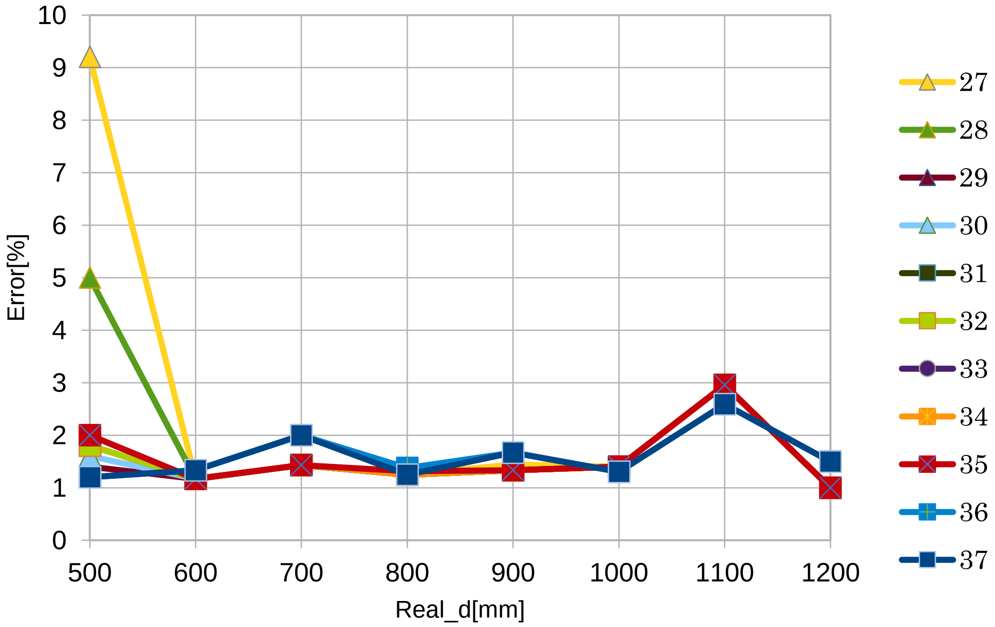


Fig. 5-23: Depth error graph for different Mindisparity.

From the result graphs, we can see that for the time consuming, the change of parameters does not improve very substantially; however, it can be seen in the depth error data that the best results can be achieved when Mindisparity is selected 29.

(2) BlockSize

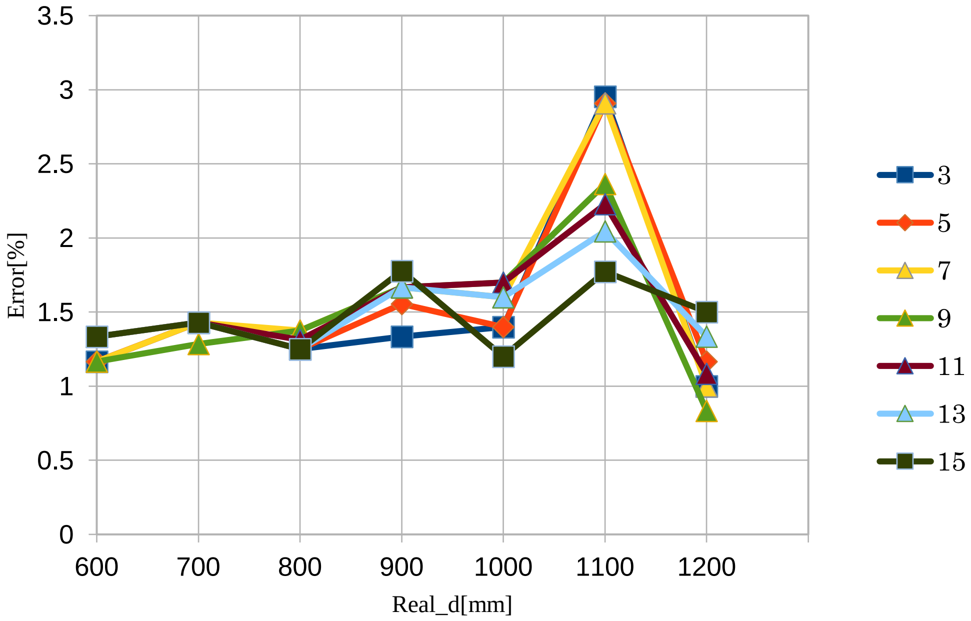
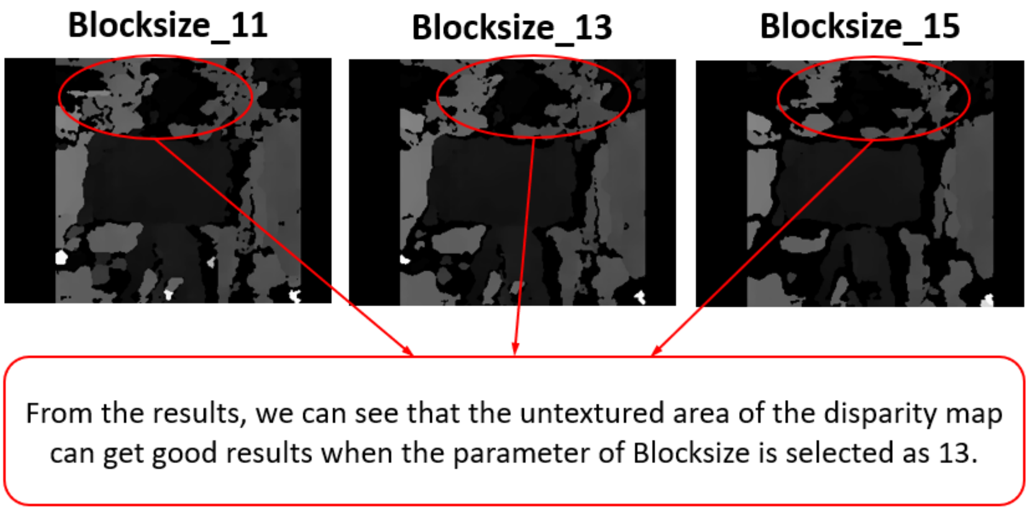


Fig. 5-24: Depth error graph for different BlockSize.



From the results, we can see that the untextured area of the disparity map can get good results when the parameter of Blocksize is selected as 13.

Fig. 5-25: Comparison of disparity map for different BlockSize.

To compare the above results, although in the depth error, the error curve is more stable, and the value is smaller when Blocksize is selected as 15. However, in the comparison of the disparity map, when Blocksize is 13, the cloud of the untextured area is better, and the black spots are smaller, the value of depth error is also more stable. Combined with the above information, so this paper selects 13 in Blocksize.

(3) NumDisparity

Table 5-10 Depth errors for different NumDisparity.

Real_d(mm)	3	4	5
500	1.4	1.4	1.4
600	1.333333	1.333333	1.333333
700	1.428571	1.428571	1.428571
800	1.25	1.25	1.25
900	1.6666667	1.6666667	1.6666667
1000	1.6	1.6	1.6
1100	2.0454545	2.0454545	2.0454545
1200	1.25	1.3333333	1.3333333

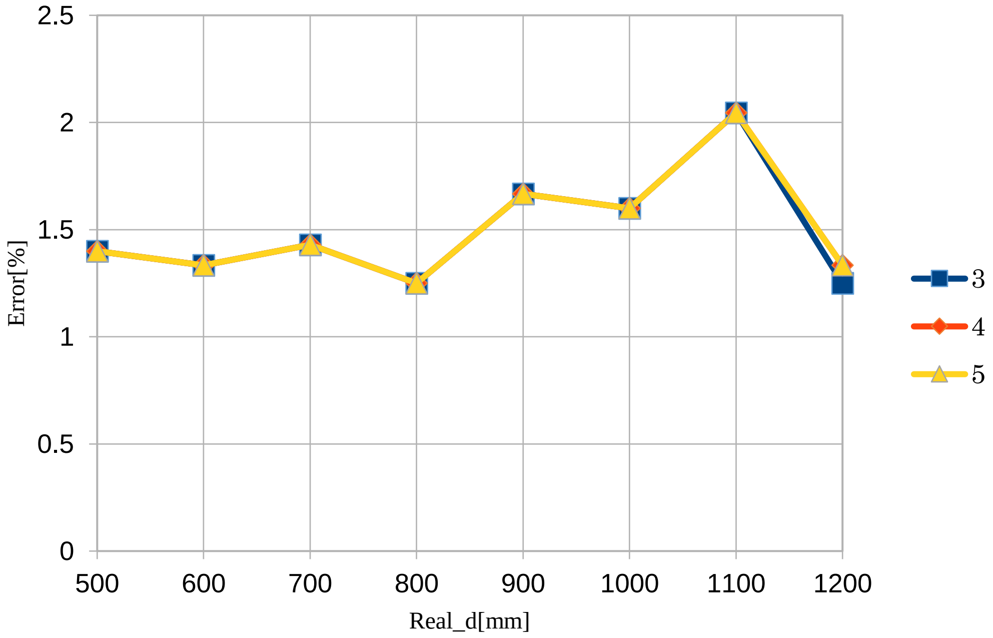


Fig. 5-26: Depth error graph for different NumDisparity.

For these data results, we can see that depth error does not change very significantly by changing the NumDisparity parameter, so in this paper, we choose 3 for the NumDisparity parameter.

(4) Mode

Table 5-11 Depth errors for different Mode.

Real_d(mm)	HH	SGBM
500	1.4	1.4
600	1.333333333	1.416666667
700	1.428571429	0.714285714
800	1.25	1.625
900	1.666666667	1.777777778
1000	1.6	1.8
1100	2.045454545	1.909090909
1200	1.25	1.583333333

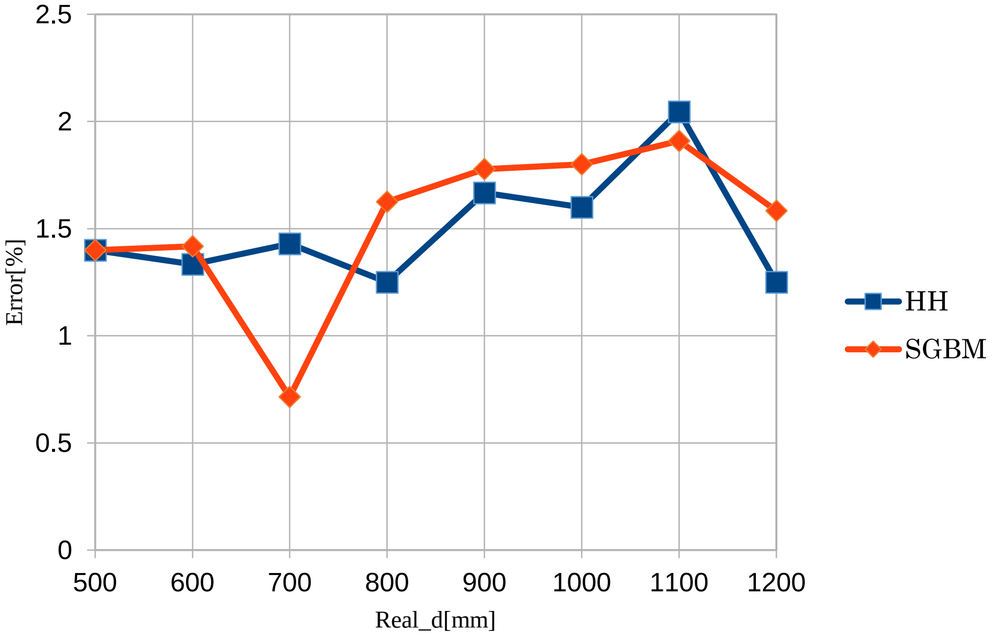


Fig. 5-27: Depth error for different Mode.

Table 5-12 Time consuming for different Mode.

Real_d(mm)	HH	SGBM
500	448.6465867	476.3508389
600	206.5670235	211.2344835
700	144.6781061	141.271524
800	198.1140311	196.6638587
900	124.5180247	126.4342964
1000	143.789551	141.6236436
1100	81.15415668	82.79590559
1200	44.51624799	43.22981143

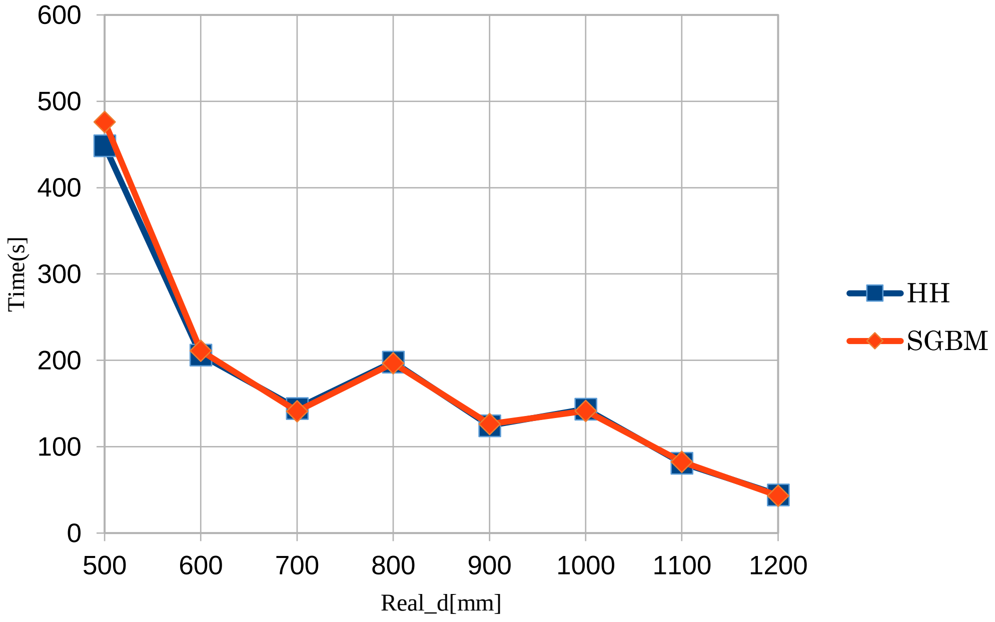


Fig. 5-28: Time consuming for different Mode.

From the above result data, we can see that the best results in terms of depth error and time consuming are obtained when Mode is chosen as HH. It should also be specified that this paper is a comparative study of the new parameters based on the previously studied parameters.

5.2. Underwater experiment

5.2.1. Experiment preparation

According to the previous section, it is known that the system has been experimented in air to get better results, and we will conduct underwater experiments next. Firstly, the binocular camera is put into a homemade pressure-resistant container, as Fig. 5-29 and Fig. 5-30 show the model picture and the physical picture of the pressure-resistant container, respectively. Because the container is to be used frequently for underwater experiments, the flange used for mounting and sealing must be anodized so that a thin protective layer is covered on the metal surface, which can improve the durability and corrosion resistance of the metal, as shown in Fig. 5-31 for the anodizing process.

In this paper, the sea urchin in the laboratory aquaculture tank (shown in Fig. 5-32) is selected, and the size of the tank is 580×290×360 (mm). Sea urchins are selected as Echinostrephus aciculatus, as shown in Fig. 5-33. According to the size of the water tank and the requirements of the conveniently moving the system, the mounting bracket is designed, as shown in Fig. 5-34.

Fig. 5-35 shows the process diagram of the experiment. Considering the thickness of the sea urchin and the necessary space of the equipment, the actual distance from the sea urchin to the camera lens is roughly 400 mm (Integer for easy calculation), as shown in Figure 5-36.

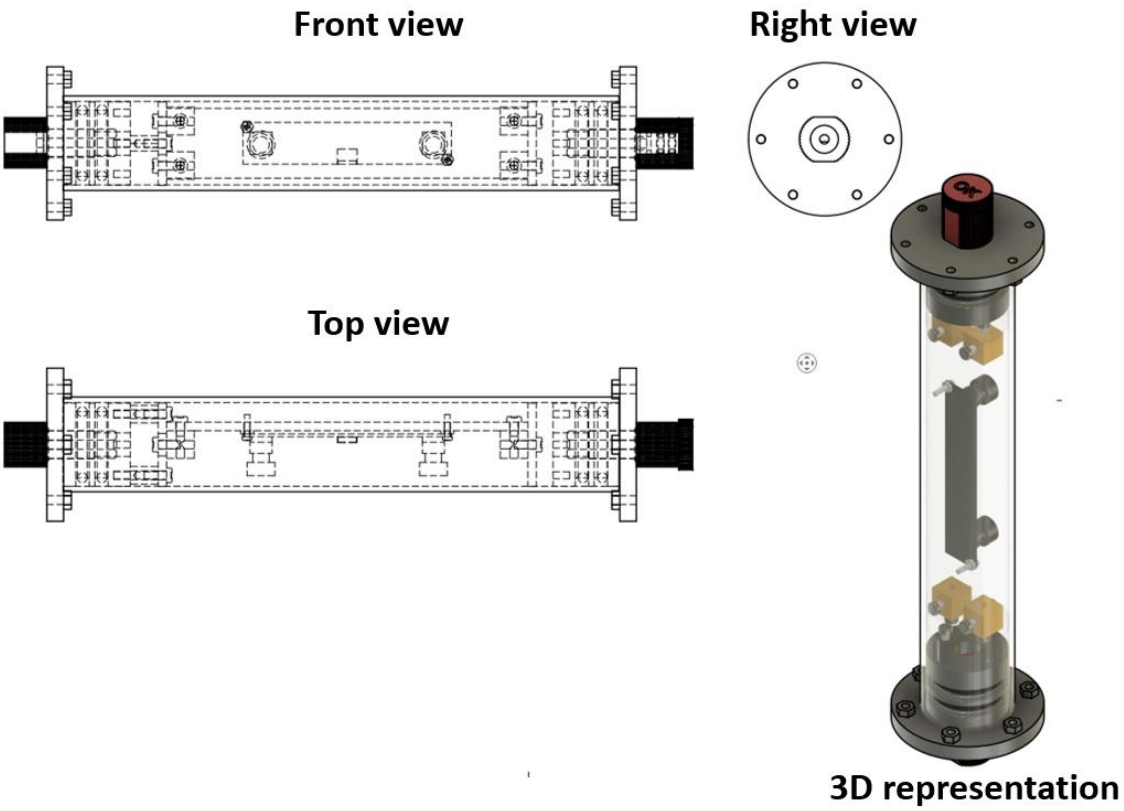


Fig. 5-29: Pressure-resistant container model.

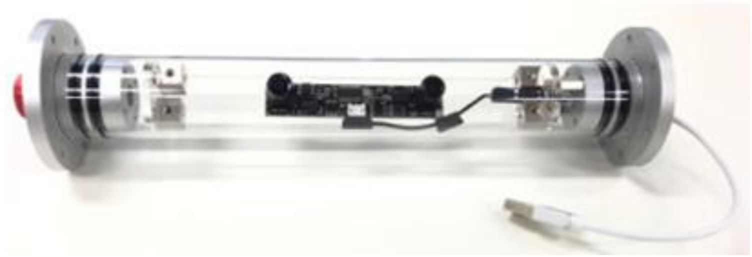


Fig. 5-30: Pressure-resistant container for underwater experiment.

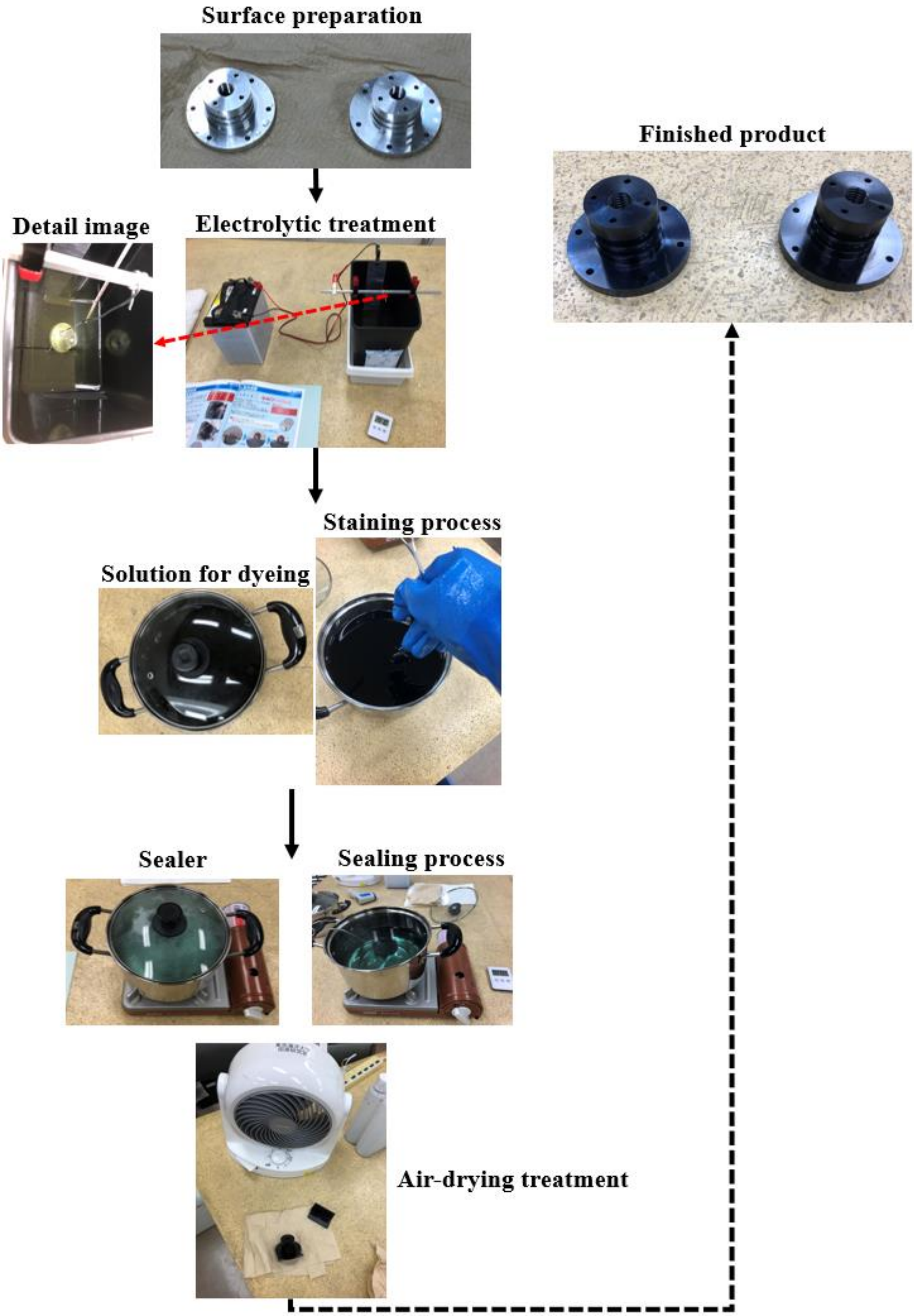


Fig. 5-31: Anodizing process.

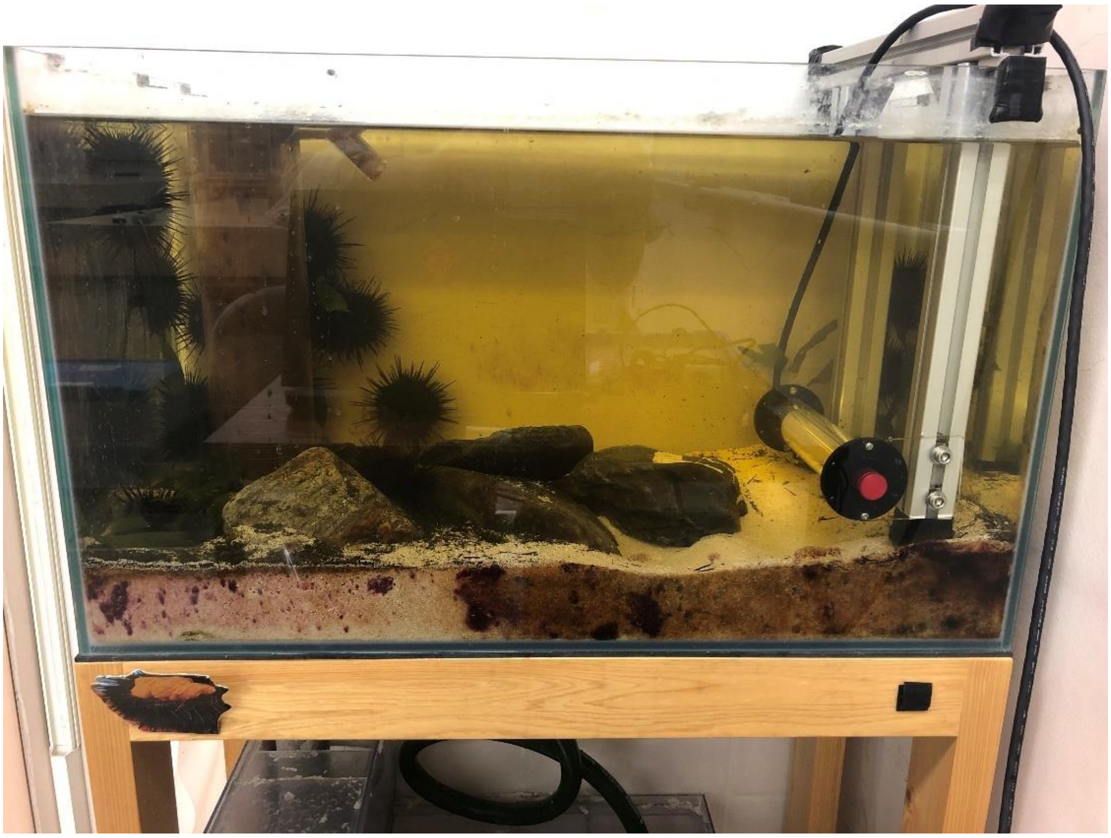


Fig. 5-32: Laboratory tank and sea urchin recognition system installation diagram.

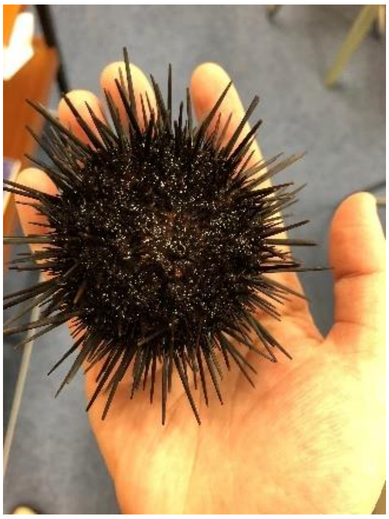


Fig. 5-33: Echinostrephus aciculatus (Tawashi sea urchin).

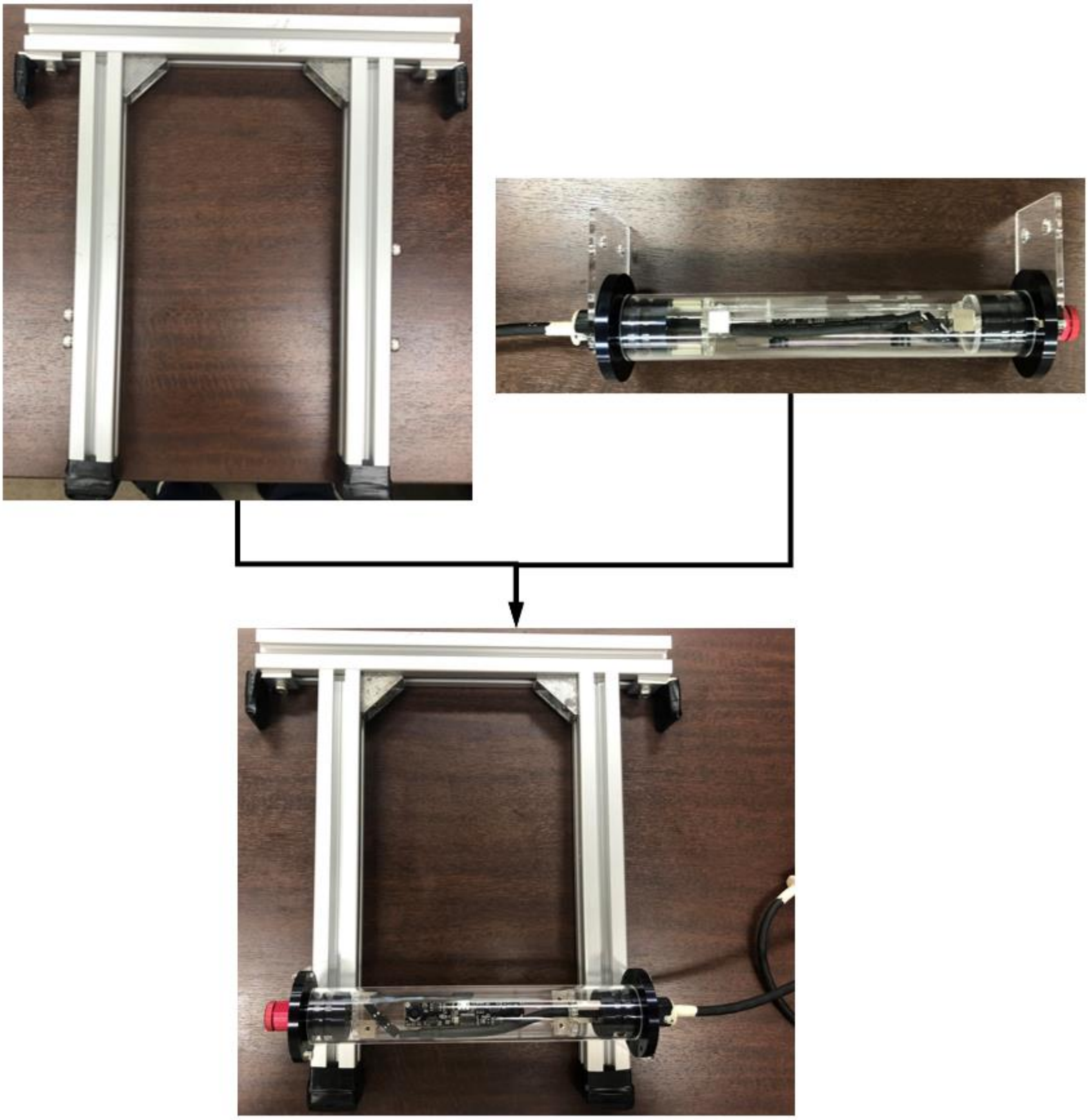


Fig. 5-34: Mounting bracket for sea urchin recognition system with underwater stereo vision.

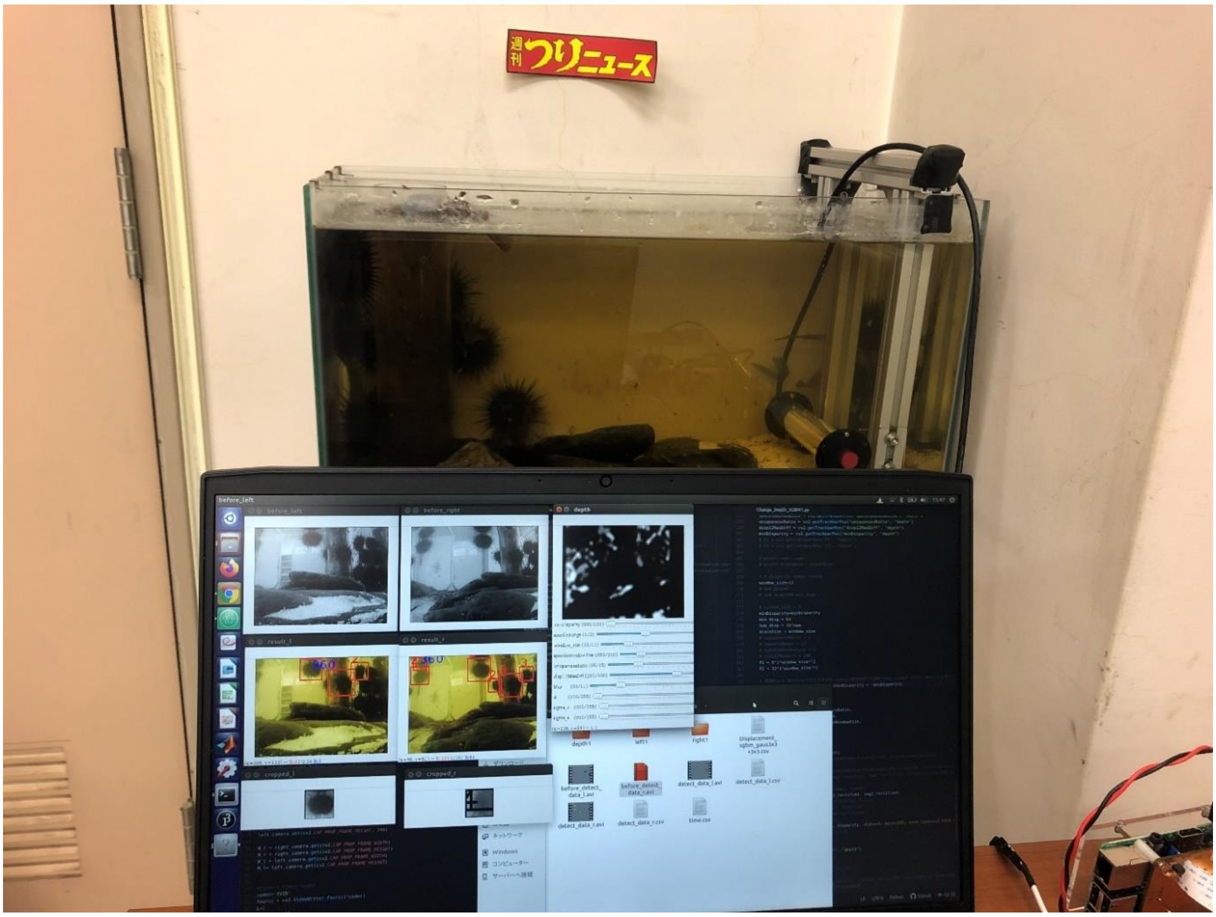


Fig. 5-35: Experimental run diagram of the sea urchin recognition system using underwater stereo vision.

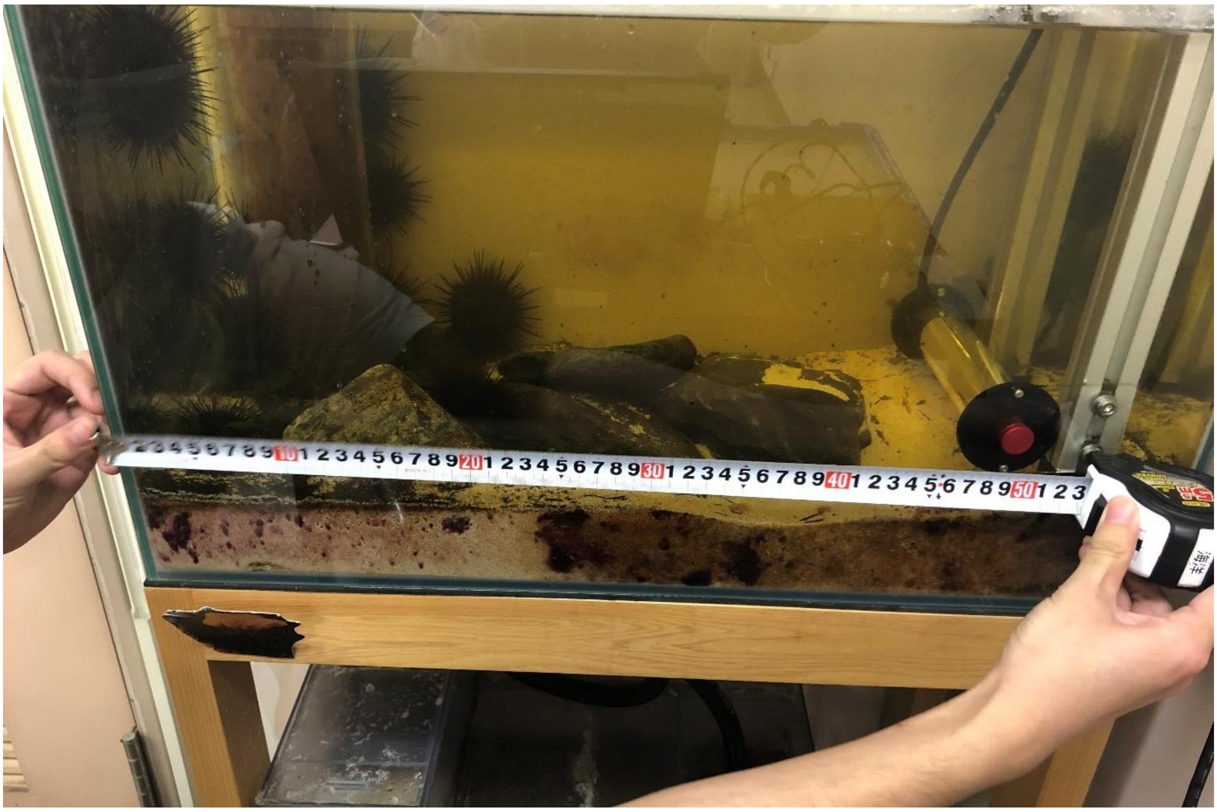


Fig. 5-36: The actual distance from the sea urchin to the system.

5.2.3. Experiment result

We used the equipment described in 5.2.1 to perform real-time recognition ranging of the sea urchins in the laboratory tank. To verify the feasibility of the code operation, this paper was tested ten times at different time periods, and the original video and the detected result video were saved each time for the subsequent research. We first performed the underwater experiments with the parameters obtained in 5.1.4, and Fig. 5-37 shows the underwater experiments result graph, where *No.* is the number of underwater experiments.

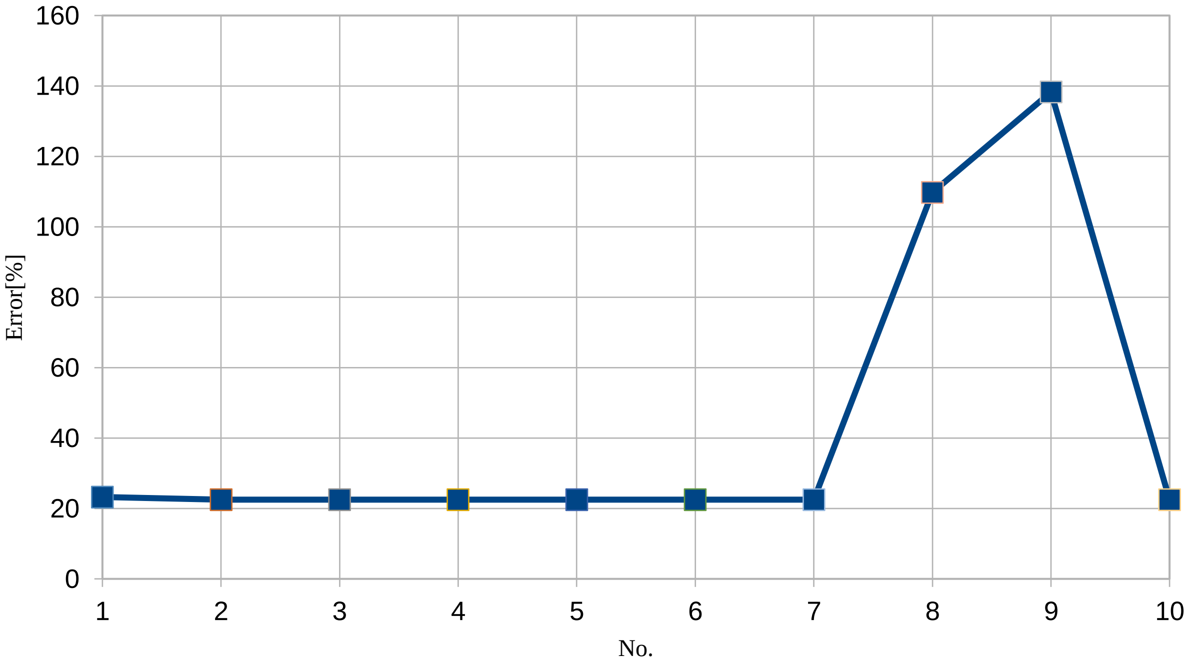


Fig. 5-37: The underwater experiments result graph.

As can be seen from Fig. 5-37 (Mindisparity=29), the stable value of the parameter obtained in 5.1.4 is between 20% and 40%, especially for two experiments with very large errors. Due to 5.1.1 obtained, the applicable range of a sea urchin recognition system using underwater stereo vision is 500 to 1200 mm (mainly depending on the camera focal length), while the maximum size of the experimental water tank is 580 mm, considering the necessary space for equipment installation and wall thickness and other factors, so the distance from the sea urchin to the camera is less than 500 mm, which can be seen from Fig. 5-36. Since Mindisparity is to change the overlapping area of left and right images by adjusting the moving the right image, Mindisparity can adjust the distance that can be detected. Fig.5-38 shows the depth error results of different Mindisparity for underwater video.

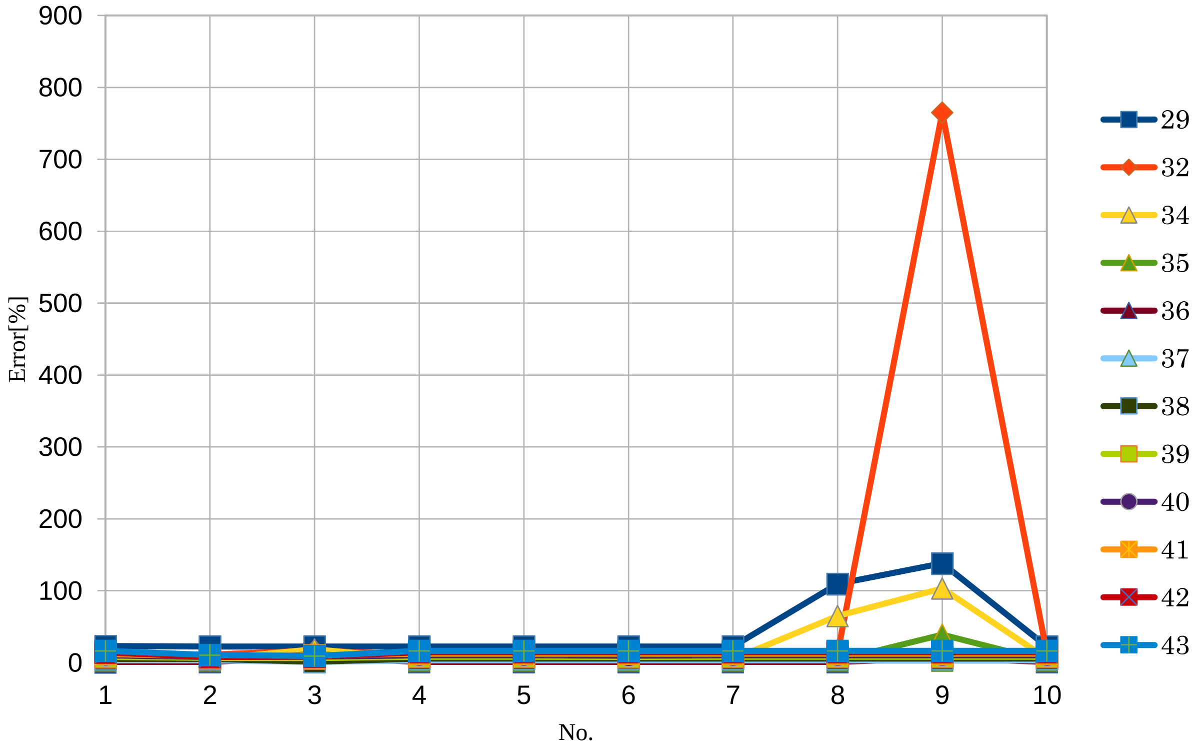


Fig. 5-38: The depth error results of different Mindisparity for underwater video.

From the results in Fig. 5-38, we can see the depth error gradually stabilizes as Mindisparity increases, especially when Mindisparity=37, the depth error of underwater detection is the smallest. Fig. 5-39 shows the results of underwater detection error when Mindisparity=37.

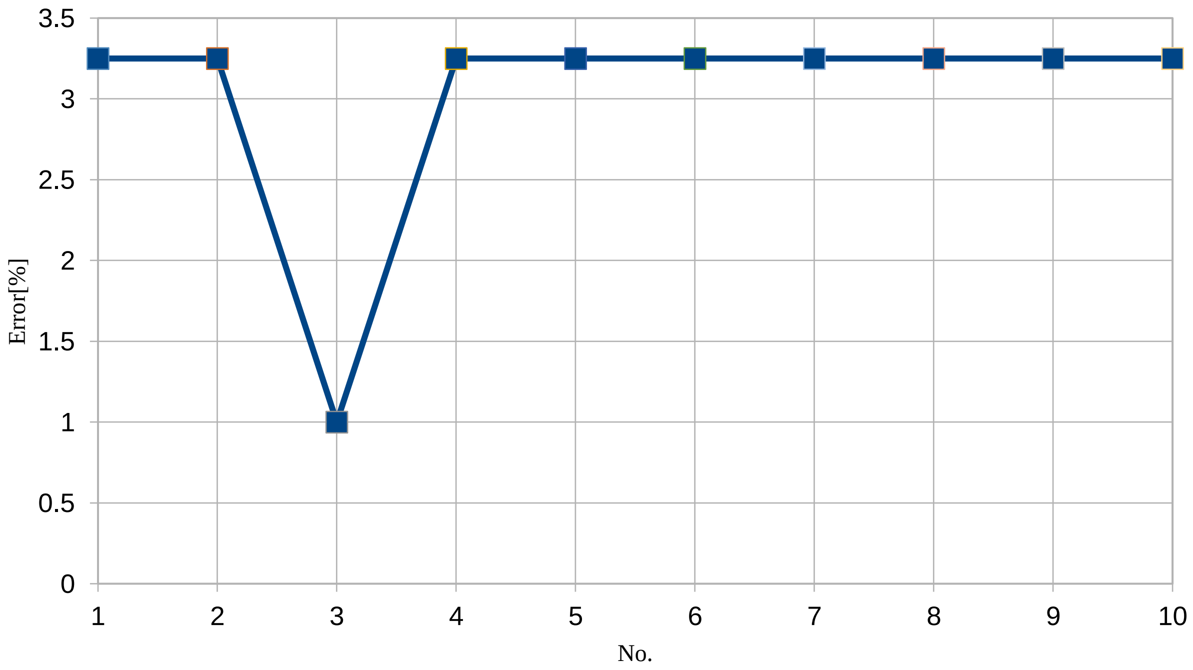


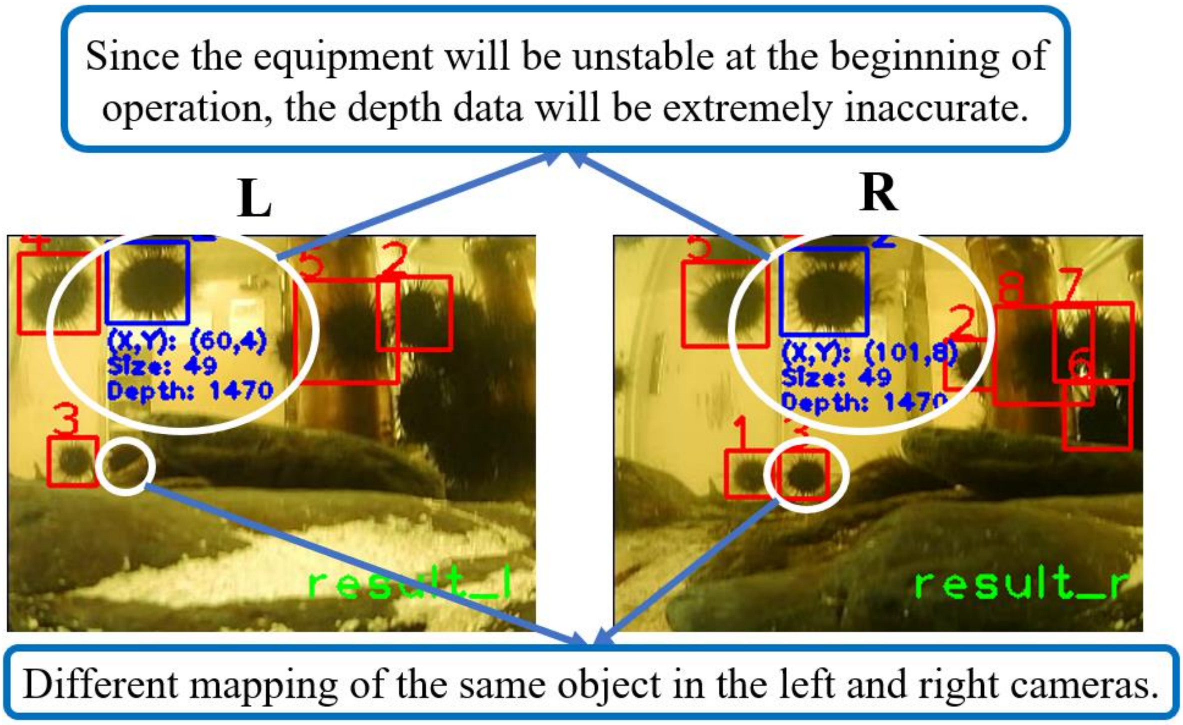
Fig. 5-39: The results of underwater detection error when Mindisparity=37

5.2.4. Analysis

The following problems can be found in the underwater experiments: 1. Depth error is particularly large and different display problems of the same object, such as Fig. 5-40; 2. Sea urchin shadow on the glass, such as Fig. 5-41. The problems will be analyzed in the next step.

For 1, since a sea urchin recognition system using underwater stereo vision is suitable for a range of 500 to 1200mm, and the aquaculture tank used in this paper has a limited size (580×290×360mm), the effect of the disparity map is not ideal. Later, a suitable camera can be used to test for different sizes of aquaculture tanks. In addition, the distance from the sea urchin to the camera is too close, which greatly worsens the different phenomenon of the same object within the left and right cameras. To solve this problem, this paper adopts reducing the similarity score to get more matching numbers of sea urchins.

At the beginning of the experiment, the system test will be unstable phenomenon of the disparity map, so there will be a large error at the beginning, but this problem will improve with time and finally the depth error value will be stable, so we can only use the experimental data of the later groups.


Since the equipment will be unstable at the beginning of operation, the depth data will be extremely inaccurate.

Different mapping of the same object in the left and right cameras.

Fig. 5-40: Depth error and different display issues for the same object.

For 2, since the glass tank used for the experiment, there will be shadows of sea urchins reflected on the glass wall, which will cause the system to misrecognize the shadows of these sea urchins. For this problem, this paper uses a white cloth hanging on the outer wall of the glass (as shown in Fig. 5-43) to make the reflection phenomenon of sea urchins improved, and the depth error data is more stable.

For the above problems, we adjusted the system by reducing the similarity value and modifying SGBM parameters (0.9 and 29 for similarity and Mindisparity in air respectively, and 0.7 and 37 in water respectively), so that the system can satisfy the requirements of both the experiment in air and water. From Fig. 5-39 and Fig.5-42, we can see the error of the experiment in air is smaller than underwater, and the constant ranging about

400mm sea urchins in water can be found that the depth error values is stable, and the values are all between 3~3.5%.

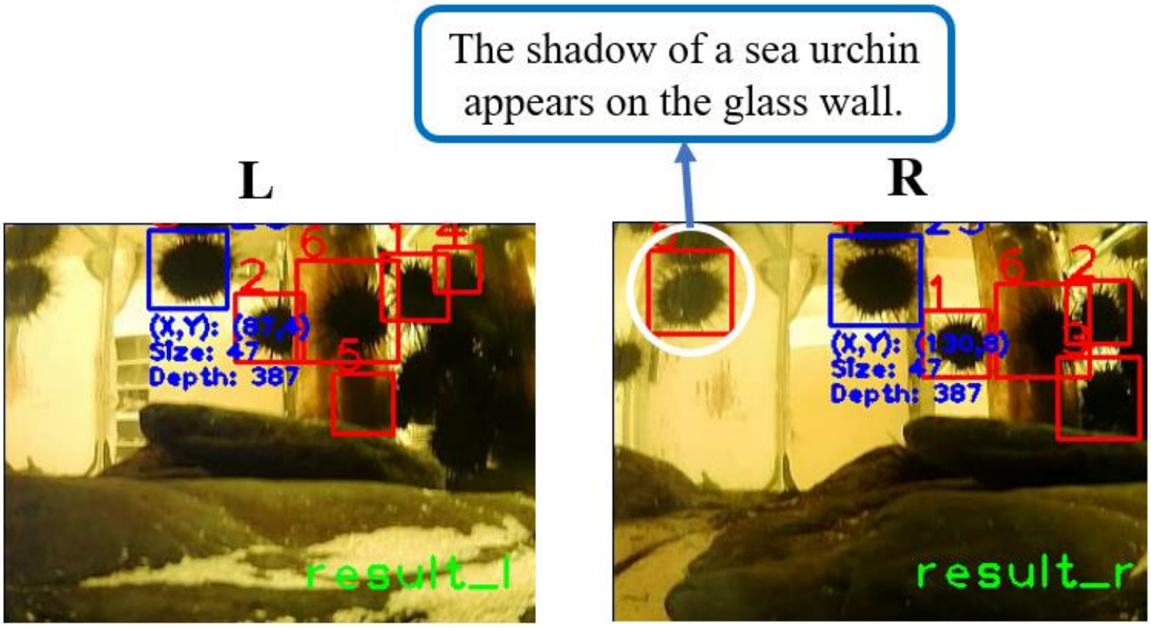

The shadow of a sea urchin appears on the glass wall.

Fig. 5-41: Sea urchin shadow on the glass problem.

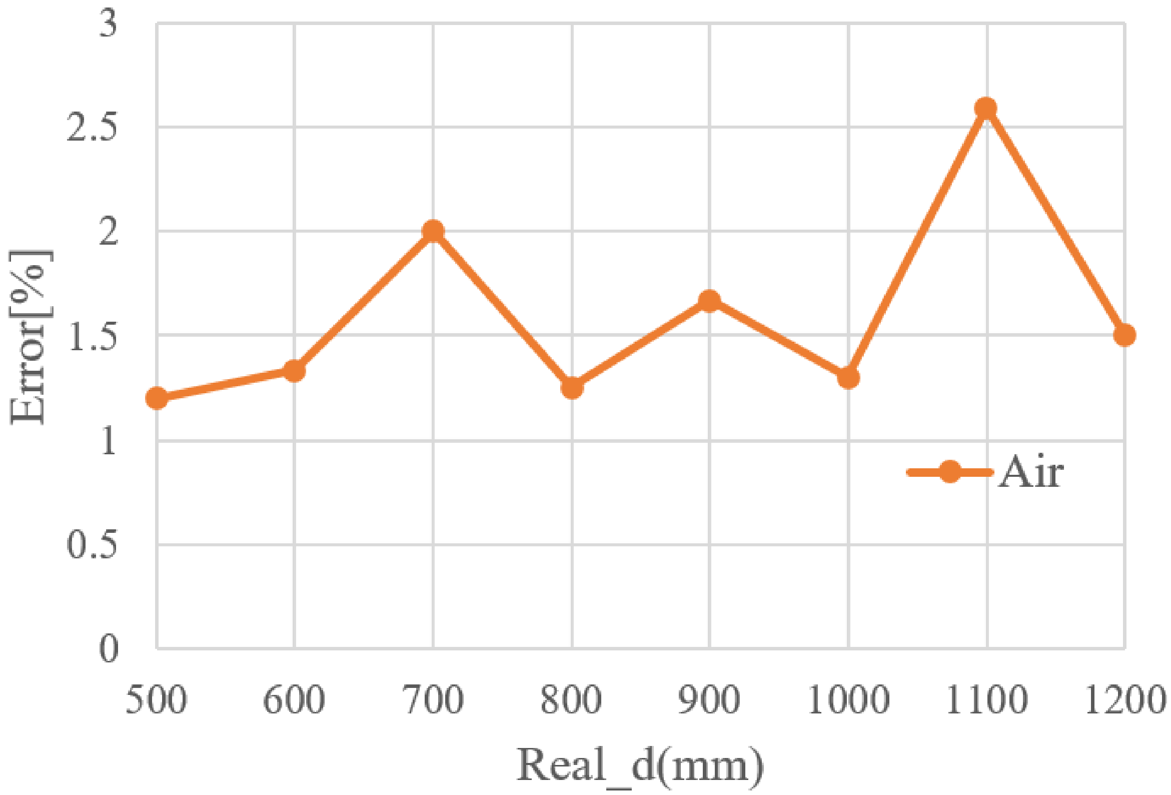


Fig. 5-42: The results of air detection error when Mindisparity=37

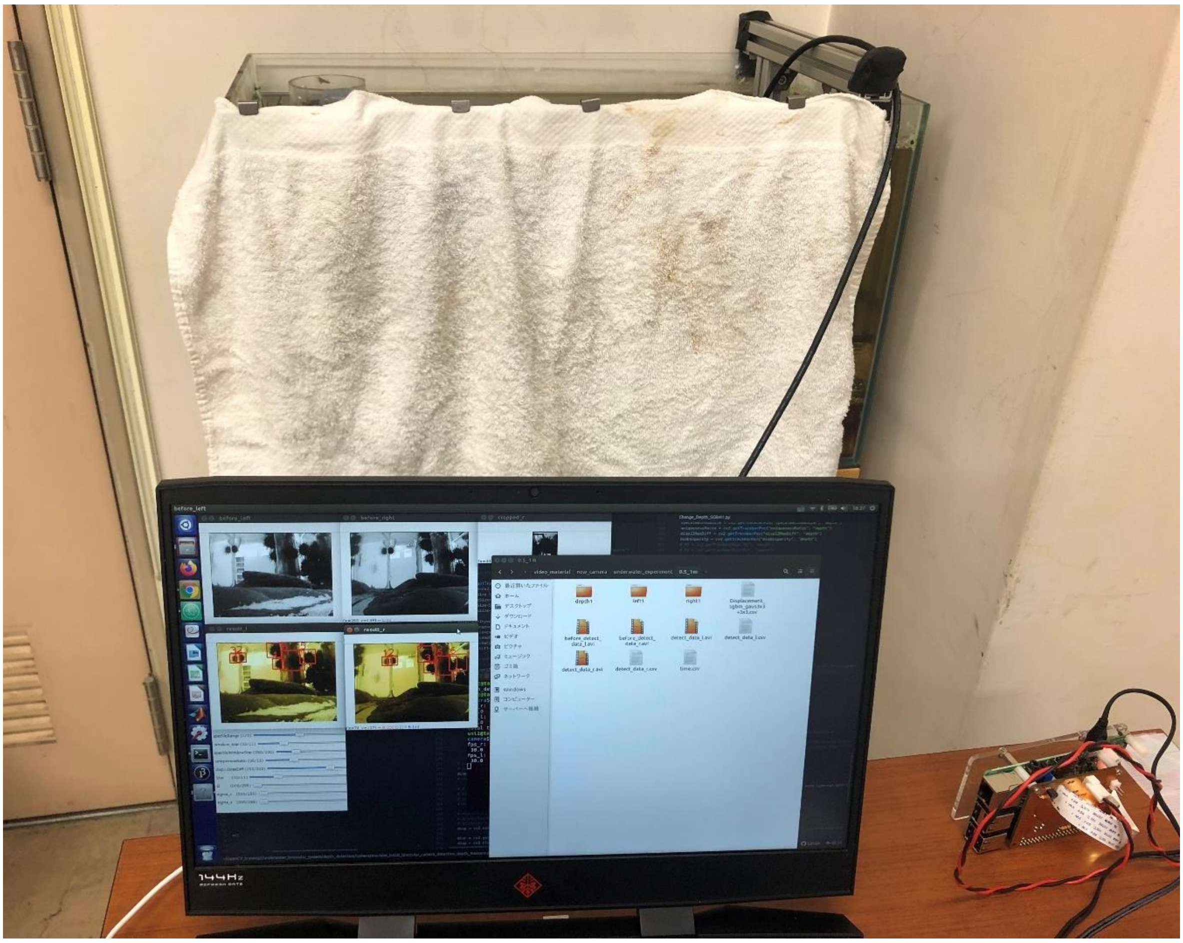


Fig. 5-43: The experimental procedure after hanging the white cloth.

6. Summary and prospect

6.1. Summary

To achieve real-time recognition of sea urchins in aquaculture tanks, so as to obtain information on the size and depth of sea urchins, this paper develops a sea urchin recognition system using underwater stereo vision. The specific work as well as the results are as follows.

(1) Binocular camera calibration and image rectification.

In this paper, we use Zhang's algorithm to calibrate the binocular camera. For the calibration results obtained by the cv2.StereoCalibrate() function in OpenCV are extremely unstable, this paper adopts the way of single camera calibration followed by binocular calibration to obtain more stable calibration results. It is found that by changing the distance and angle to take at least 80 calibration photos, better calibration parameters can be obtained. Through analysis and comparison, when the translation matrix obtained from the experiment is close to the distance of 62mm between the two cameras (Ideally the translation matrix should be $[62,0,0]$), as shown in Table 6-1, the translation matrix of the three sets of better parameters obtained from the calibration experiment.

We get the error of the translation matrix with the ideal state of the translation matrix ($[62,0,0]$) by using Eq. 6-1 and get the error table of the translation matrix in Table 6-2 and the error diagram of different parameters shown in Fig. 6-1.

$$Error = \frac{|Tx - 62|}{62} \times 100\% \tag{6-1}$$

Table 6-1 Calibration parameters (translation matrix)

No.	T_x	T_y	T_z
A	61.766	-0.69659	-13.8499
B	62.0442	-3.55082	-20.6148
C	62.333	-3.726065	-14.2267

Table 6-2 Translation matrix error table

No.	Error_x	Error_y	Error_y
A	0.234	0.696587	13.84993
B	0.0442	3.550818	20.6148
C	0.3327	3.726065	14.22668

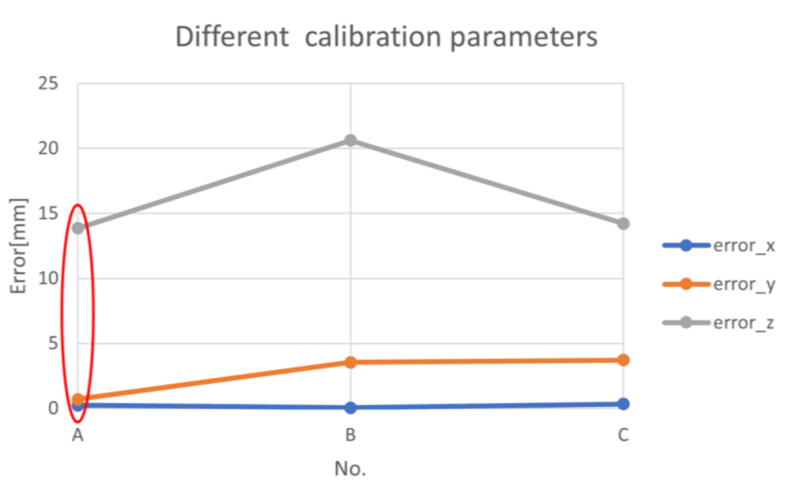


Fig. 6-1: Translation matrix error result graph

The above results show that the parameter translation matrix of group A is the closest to the ideal state, which is also consistent with the previous experimental comparison results. Therefore, it is considered necessary to judge the merits of the calibration parameters with the results of the translation parameters as the benchmark in the future.

(2) Image processing and sea urchin recognition

It is found that grayscale processing, histogram equalization, and image filtering of the image are beneficial to improve the detection accuracy. According to the comparison of the experimental data, it is necessary to perform bilateral filter on the image before image correction, and the optimal parameter is chosen as 3. After generating the disparity map, it is necessary to perform blur filter on the newly generated disparity map, and the optimal parameter is chosen as 10.

(3) Similarity calculation and stereo matching

Since this paper uses the sea urchin classifier previously developed in the laboratory to recognize sea urchins, it is different from the general binocular ranging sequence. In this paper, we first let the left and right cameras recognize the sea urchins, and then perform similarity calculation on the detection frames obtained by the left and right cameras on this basis, so as to get the pixel coordinate values of the sea urchins detected by the left and right cameras at the same time. The effect diagram and flow chart of the sea urchin recognition system using underwater stereo vision developed in this paper are shown in Fig. 6-2 and Fig. 6-3, respectively.

For the similarity calculation, this paper firstly uses machine learning or deep learning to get the similarity value of two patches. Since machine learning is less time consuming and has good depth error results, the machine learning method (Template matching) is chosen in this paper.

We obtain the disparity value of the sea urchin by using the coordinate values of the sea urchin detected by the left and right cameras simultaneously and the disparity map obtained by stereo matching using the optimal parameters, so that the distance from the sea urchin to the binocular camera can be obtained by the calculation formula.

(3) Experimental results

In this paper, experiments were first conducted in air, and a better set of system combinations and parameters were obtained through continuous testing of sea urchin pictures. Based on this, underwater real sea urchin experiments were then conducted. By making Mindisparity=37, a sea urchin recognition system using underwater stereo vision was made to suit both air and the aquaculture tank in this lab (this can be seen in Fig. 5-23 and Fig. 5-39). The sea urchin results graph is shown in Fig. 6-4.

For the above problems, we adjusted the system by reducing the similarity value and modifying SGBM parameters (0.9 and 29 for similarity and Mindisparity in air respectively, and 0.7 and 37 in water respectively), so that the system can satisfy the requirements of both the experiment in air and water. From Fig. 10 and Fig. 11, we can see the error of the experiment in air is smaller than underwater, and the constant ranging about 400mm sea urchins in water can be found that the depth error values is stable, and the values are all between 3~3.5%.

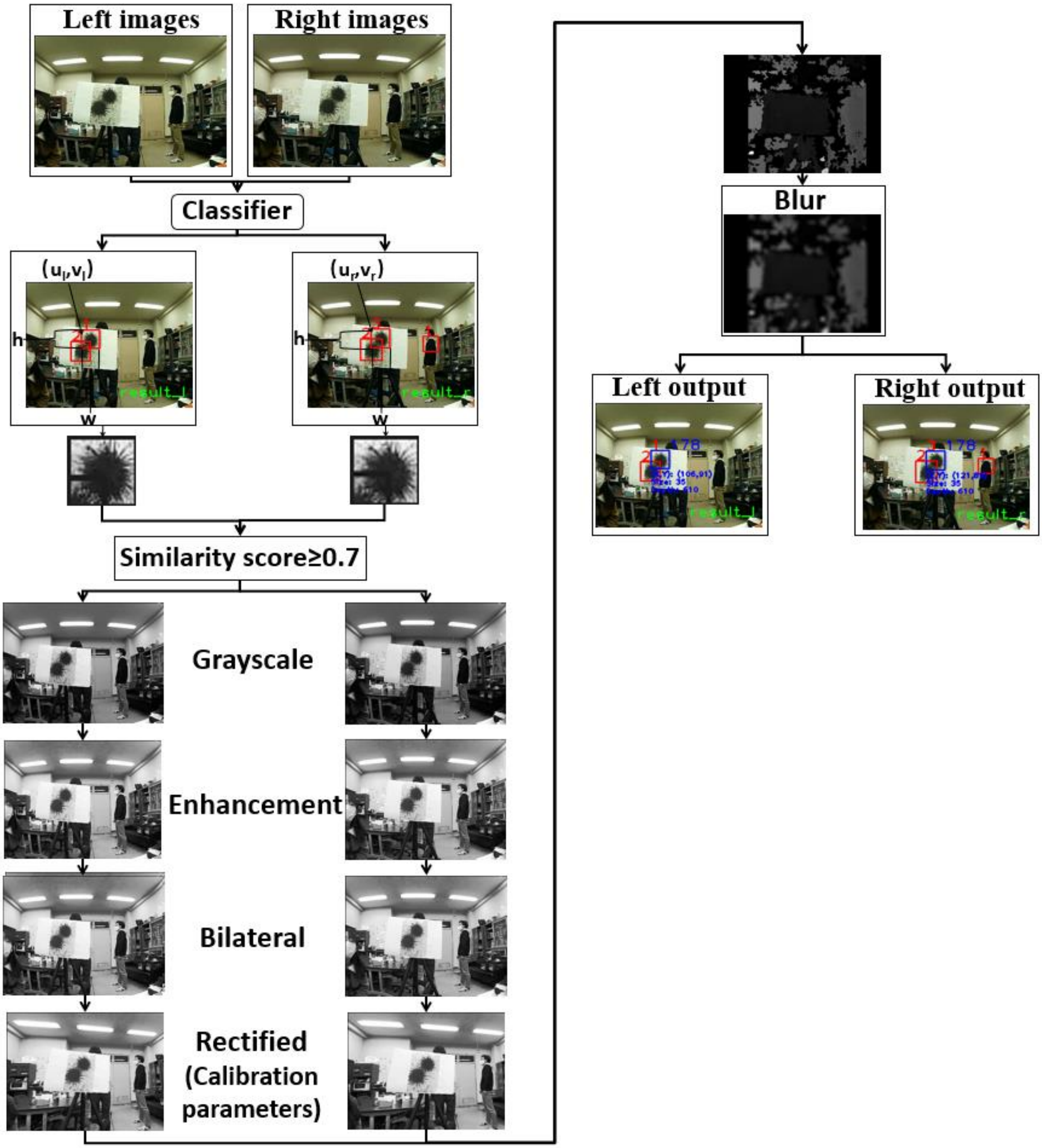


Fig. 6-2: The effect diagram of sea urchin recognition system using underwater stereo vision.

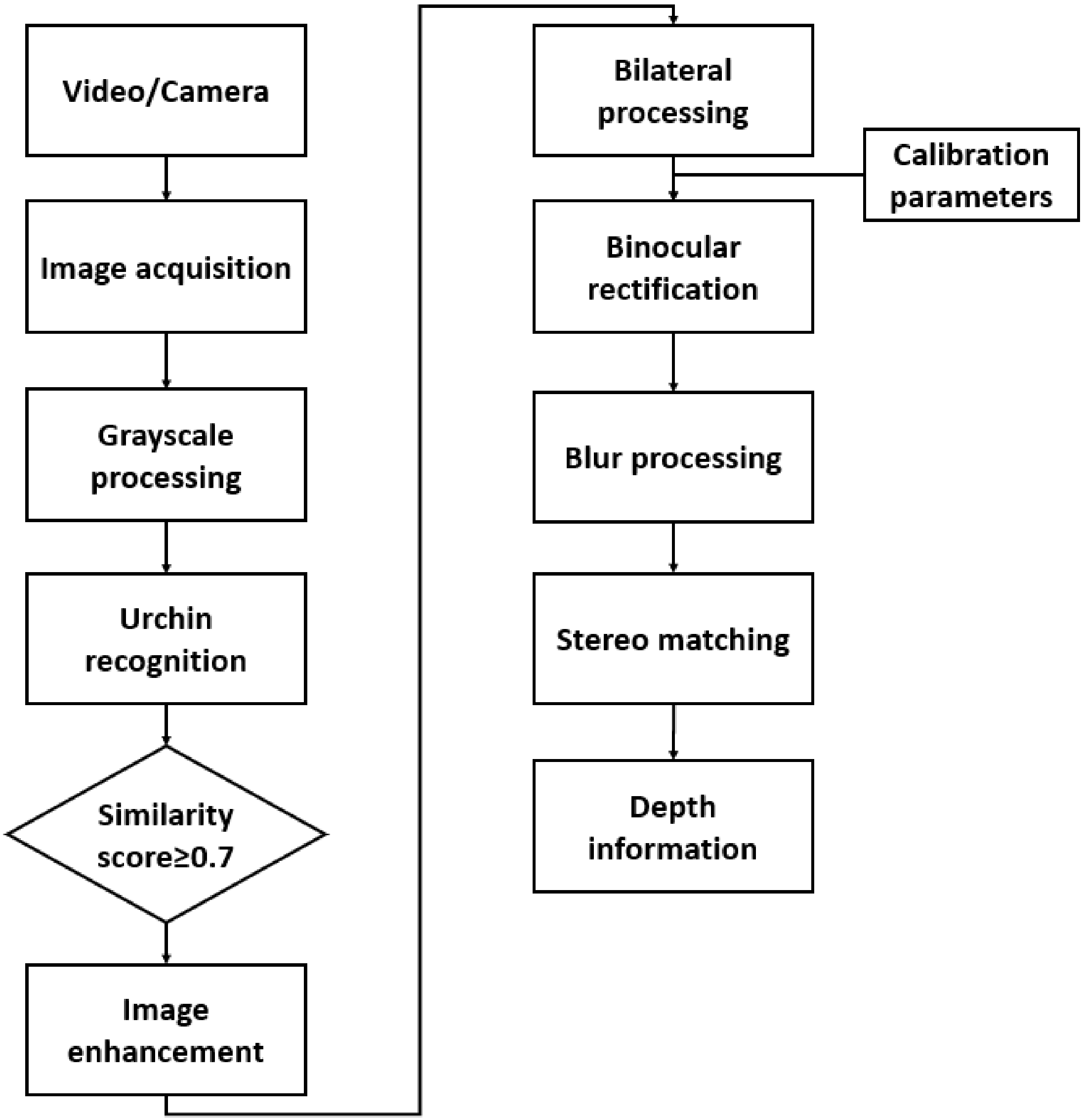


Fig. 6-3: The flow chart of sea urchin recognition system using underwater stereo vision.

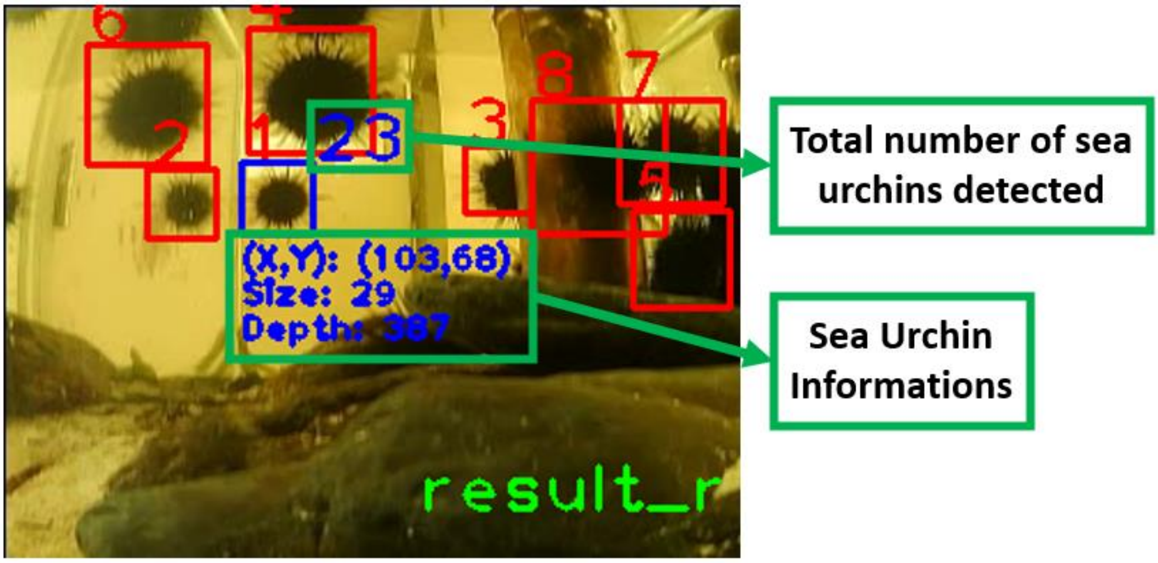


Fig. 6-4: Illustrative diagram of system operation effect information.

6.2. Prospect

This paper studies a sea urchin recognition system using underwater stereo vision applicable to aquaculture farming, which is small in size, light in weight and low in cost, suitable for the aquaculture industry. In the future, it is possible to construct an IoT underwater camera system for aquaculture by combining a simple conductivity sensor and water thermometer.

Reference

[1] 川村大和，田原淳一郎，和泉充，井田徹哉，阿部拓三，“小型 ROV を用いた生物調査システム”，海洋理工学会平成 30 年度春季大会, (5.2018) pp39-40.

[2] 加藤哲，川村大和，田原淳一郎，和泉充，井田徹哉，阿部拓三，"小型 ROV システムの動作特性"，海洋理工学会平成 30 年度春季大会, (5.2018) pp65-68.

[3] 伊藤魁，後藤慎平，和泉充，井田徹哉，田原淳一郎，"ウニ回収 ROV の回収システムの概要と実地試験結果"，海洋調査技術学会第 31 回研究成果発表会, (11.2019) pp25-26.

[4] 川村大和, SonMunseong，齋藤幹大，伊藤魁，加藤哲，田原淳一郎，和泉充，“ROV を用いたウニ駆除システム”，海洋理工学会 2019 年秋季大会, (11.2019).

[5] 齋藤幹大, "小型 ROV に搭載するウニ認識システムの開発", (2020.3).

[6] 生態系を脅かしかねない増え過ぎた宮城県・志津川湾のウニ ｜ ソーシャル・イノベーション・ニュース（social-innovation-news.jp).

[7] 齋藤幹大，田原淳一郎，加藤哲，川村大和，"機械学習手法を用いた画像認識システムの開発"，ロボティクス・メカトロニクス学会, (6.2019) p29.

[8] 陈莉, "基于机器视觉的双摄像头测距系统的研究与实现." 河北科技大学. 2019.

[9] A. Kaehler, G. Bradski, "Learning OpenCV 3: computer vision in C++ with the OpenCV library", O’Reilly Media Inc, (2016) 644-648.

[10] Zhang, Z., "A flexible new technique for camera calibration", IEEE Transactions on Pattern Analysis and Machine Intelligence, 22 (2000) 1330-1334.

[11] OpenCV: Smoothing Images.

[12] Viola, P., Jones, M., ""Rapid object detection using a boosted cascade of simple features"." I-I. 2001.

[13] Contrastive Loss for Siamese Networks with Keras and TensorFlow - PyImageSearch.

[14] H.Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information", IEEE Transactions on Pattern Analysis and Machine Intelligence, 30 (2008) 328-341.

[15] OpenCV: cv::StereoSGBM Class Reference.

[16] Chai Jiayu, Junichiro Tahara, "Underwater Distance Measurement Using Machine Learning", Proceedings of the 2021 JSME Conference on Robotics and Mechatronics, No. 21-2 (6.2021).

Acknowledgement

Time flies. In this rainy season, I will finish my graduation thesis and my two-year graduate career will come to an end. Looking back on the past, it is a great honor for me to spend my best years in such a campus with many excellent teachers and students.

This paper is completed under the kind care and careful guidance of Professor Junichiro Tahara. His rigorous scientific attitude, working style of striving for perfection, his noble teacher's morality, his lofty demeanor of being strict with himself and being lenient to others, and his simple and approachable personality charm deeply affected and inspired me. From the selection of the project to the final completion of the project, Prof.Tahara has always given me careful guidance and unremitting support. Every time, he has given me careful answers to my questions and given writing suggestions. He has carefully revised my paper, which has made my thesis structure gradually improved and the content has become more and more abundant. Without Prof.Tahara 's careful guidance, this paper is impossible to complete. I would like to express my sincere thanks and high respect to Prof.Tahara.

I am grateful to Prof. Zhang, Prof. Koike, and other teachers at Tokyo University of Marine Science and Technology, whose rigorous academic attitude and humorous teaching methods have left a deep impact on me and provided me with a great deal of material and profound insights for the writing of this paper. I am grateful to my partners in the laboratory, Mr. Fujii, Mr. Ono, and Mr. Morito, for helping me with experiments and providing me with literature so that I could have a reference point in writing this paper. Thank you to my classmates for sharing these two happy and fulfilling years with me, and for your support and help in my regular work. I would like to thank my former and current colleagues who worked with me for their help in the process of writing my dissertation.

Finally, I would like to thank my parents deeply. It is their silent support and selfless love that have enabled me to have infinite motivation on the road of study and enjoy warm dependence in the journey of life. You will always be my favorite person.

In the process of writing this paper, I feel that my level is still very poor. I will not take the end of this paper as an end, but it will be a new starting point of my life.

There is no end to learning. I am willing to forge ahead in the future!