Fishery monitoring system with AUV based on YOLO and SGBM

Haotian Wu[1], Shimin He[1], Zejun Deng[1], Liwei Kou[1], Kangwei Huang[1], Feiyang Suo[1], ZheCao[2]
1. Department of Electrical and Electronic Engineering, Zhejiang University, Hangzhou 310000, China
2. Department of Mechanical Engineering, Zhejiang University, Hangzhou 310000, China
E-mail: haotianwu@zju.edu.cn

Abstract: AUV has gained more and more attention with the growing importance of the ocean. The visual intelligence extends the sensing ability of the AUV. Prior work on visual intelligence of AUV mainly focuses on the object classification and serving areas underwater. Instead, we focus on a better understanding of images and get more useful information from binocular cameras. This paper introduces a new fisheries application of the visual intelligence of AUV and proposes a new approach to monitor the growth of fish with binocular cameras. We construct a deep network with YOLO architecture to detect the bounding boxes of fishes and extract the edges of fishes from the bounding boxes.Then we use the SGBM method to predict the depth map, which is used to estimate the length and width of the fishes. The YOLO object detection system and SGBM algorithm are both low-cost with accurate real-time results in experiments. This fishery monitoring system is efficient, real-time and accurate compared with the manual method or dual-frequency sonar.

Key Words: Binocular monitoring, AUV, Fishery application, Fish size estimation, Stereo matching, Fish detection

1 Introduction

AUV (autonomous underwater vehicles) is generally designed for many marine applications, such as ocean exploration and underwater operation. But there is very limited information that AUV can access for the rough underwater environment (without effective communication or navigation feedback) [1]. Sonar is a typically expensive measurement, but it only can get the geometry construction information which is not enough for many underwater operations and monitoring tasks [2].

One of the most critical applications of AUV this paper concern is the fishery. For the fishery application, the growth of the fishes (estimated by the length or weight of the fishes) is one of the most important parameters to show whether the fish grows in a good situation. In the past, this parameter is measured by a manual process where people collect the fishes from the water and measure or weigh them before putting fishes back. But AUV shows the potential of replacing humanity to complete this repeated task. The dual-frequency identification sonar acoustic camera is a typical method to measure the size of fish directly [3], but the high cost of the dual-frequency identification sonar is the main disadvantage of this method.

Over the past decades, visual intelligence has some exciting technological improvements for the advance of the computation capability. Many computer vision algorithms have been developed for target tracking and object detection, which is very meaningful for AUVs to compensate for its limited information underwater. Computer vision has also been an effective method to extend the AUV's sensing capability with low cost, which gains lots of success in many parts. Visual intelligence can help AUV compute the relative displacement from the specific scene features[4]. It also makes sense in the visual inspection and serving areas[5] [6].

In this paper, we introduce a new visual system for the AUV and propose a new method to monitor the growth of fish. We firstly calibrate the binocular camera with the method from the paper [7] to get the intrinsic and extrinsic parameters, which are useful for the later stages. Then we construct the deep network with YOLO architecture[8] to detect the fishes in the image and use the 'Stereo Processing' method (SGBM algorithm) to compute the disparity map, which is used to predict the depth map of the 2-D images[9]. With the depth map and the bounding boxes (the result of the YOLO network), then we can estimate the size of fishes from the similar triangle principle.

This fishery monitoring system has many advantages. Binocular cameras have more accurate performances in the estimation of the depth information than the monocular camera. YOLO architecture has the benefits of real-time detection which is very accurate and low-cost. SGBM (Semi-global block matching) is one of the most efficient algorithms to compute the disparity. So this YOLO and SGBM based binocular monitoring system is accurate, efficient and low-cost compared with the dual-frequency identification sonar or manual method.

In this paper, we first demonstrate the proposed method and the construction of this fishery motoring system. We explain this system from the camera calibration underwater, YOLO fish detection and SGBM algorithm. And then we do some experiment to show some advantages of this system. In the last, we also point some potential improvement of this system in the future.

2 Proposed method

In this section, we demonstrate a AUV based fishery monitoring system (Figure 1) which consists of AUV platform, camera calibration, YOLO fish detection, SGBM disparity computation and the size estimation of fishes.

As the motion of fish is normally random, we could use the AUV with the visual intelligence to monitor the situation of fish. We add the visual part in the front of AUV platform and make the AUV move randomly underwater to make sure that it could monitor all the fishes.

This work is supported by the Key R & D Program of Zhejiang Province (2019C02002) and the Fundamental Research Funds for the Central Universities (2018XZZX001-06).

For the input images of the system, this fishery monitoring system detects the fishes from images and gets the bounding box with confidence scores. For the images in the bounding boxes, we extract the edges of fishes with binarization process and Canny operator. And we keep the edges with the most continuous pixels as the fishes. With the intrinsic and extrinsic parameters from camera calibration, the system computes the disparity and depth map of the images in the bounding boxes by SGBM method. Given the depth information and camera parameters, we can get the real world distance of any two pixels in the image. So we can use this method to estimate the size of the fish.

2.1 AUV Platform

The AUV Platform we used is QLAUV-II (quadrotor-like autonomous underwater vehicle), developed by Kou [10]. QLAUV-II is in cylindrical shape and its two sides are both elliptical, where we place the camera.

Figure 2 shows the shape of the QLAUV-II. Figure 3 shows the mechanical drawing of QLAUV-II. QLAUV-II has four thrusters which are placed symmetrically in a 'X-shape' at two sides of the AUV. This QLAUV benefits for its independent horizontal and vertical motion, which makes it a useful tool for our project to monitor the fishes underwater. And the lighting device is set in the front of the AUV.

2.2 Calibration

For a binocular computer vision task, the first thing to do is the calibration. With the calibration, we can get the intrinsic and extrinsic parameters to predict the depth map.

Using the method from the paper[7], we denote the pixel coordinate: (u, v), image coordinate (x, y), camera coordinate (X_c, Y_c, Z_c) and world coordinate (X_w, Y_w, Z_w). From the principle of keyhole imaging and similar triangles:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a & r & u_0 \\ 0 & b & v_0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

$$Z_c * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{3}$$

Where a, b is the scale relationship between pixel and image coordinate. r is the distortion parameter, which can be ignored in common situation. (u_0, v_0) is the origin of the image coordinate. f_x and f_y are focuses of the camera. R is the rotation matrix, T is the translation matrix.

So we can express the relationship between the pixel coordinate and world coordinate with intrinsic parameter matrix M_1 and extrinsic parameter matrix M_2:

$$\begin{aligned} Z_c * \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} a & r & u_0 \\ 0 & b & v_0 \\ 0 & 0 & 1 \end{bmatrix} * \\ &\begin{bmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\ &= M_1 * \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\ &= M_1 * M2 * \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \end{aligned} \tag{4}$$

To adapt to the underwater environment, we need to calibrate the cameras underwater, where we use the Zhang Zhenyou calibration method[7] for its efficient performance. The basic equations of this method are:

$$\begin{aligned} s * \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= M_1 * M_2 * \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\ &= M_1 * [R_{3\times3}, T_{3\times1}] * \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \end{aligned} \tag{5}$$

Where s is an arbitrary scale factor, $[R_{3\times3}, T_{3\times1}]$ is the extrinsic parameters matrix(rotation and translation), M_1 is camera intrinsic parameters matrix (scale related).

To solve this problem, we define the $\widetilde{m} = [u, v, 1]^T$, $\widetilde{M} = [X_w, Y_w, Z_w, 1]^T$, $s\widetilde{m} = A[R \quad t]\widetilde{M}$

$$A = M_1 = \begin{bmatrix} \alpha & r & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, R_{3\times3} = [r_1, r_2, r_3] \tag{6}$$

Set $Z_w = 0$ and define $H = A[r_1 \quad r_2 \quad t] = [h_1, h_2, h_3]$:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A[r_1 \quad r_2 \quad t] = \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \tag{7}$$

With the relation $[h_1, h_2, h_3] = \lambda A[r_1, r_2, t]$, $r_1^T r_2 = 0$, $r_1^T r_1 = r_2^T r_2 = 1$, we can get A from H:

$$\begin{aligned} &h_1^T A^{-T} A^{-1} h_2 = 0 \\ &h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2 \end{aligned} \tag{8}$$

The matrix $H_{3\times3}$ has eight unknowns and each coordinate can get two equations. Matrix A has five unknowns, so we need three matrices to compute the matrix A, which leads to three different pictures underwater. So we need to take at least four pictures underwater for calibration.

2.3 YOLO based fish detection system

The visual intelligence of the AUV aims to detect the object underwater. The network we use is 32-layer YOLO architecture (YOLO v2) for its good performance in mAP and

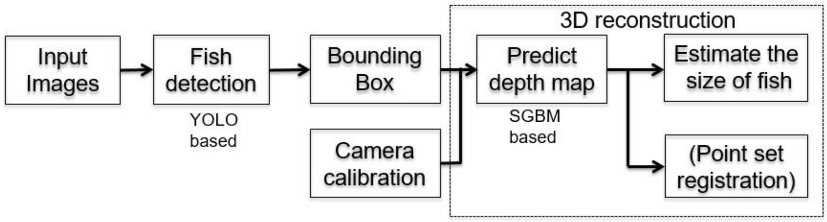

Fig. 1: The fishery monitoring systems

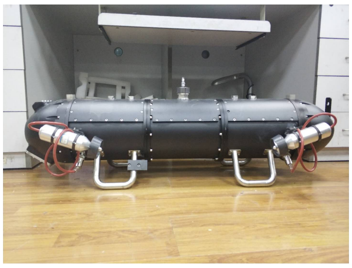

Fig. 2: The real-shape of the QLAUV-II

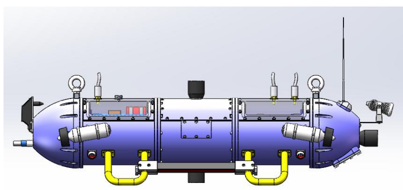

Fig. 3: The mechanical drawing of QLAUV-II

FPS value. YOLO algorithm could run the detection system in real-time on Titan X GPU[8] (45 frames per second at least).

The basic pipeline of the YOLO detection systems is that YOLO divided the image into $S \times S$ grid and predict the bounding box of the object with each grid cell. Since the original YOLO is evaluated on PASCAL VOC detection dataset, where the scale of fish is large and the environment is different for the aim of generalization, the original YOLO is not accurate and applied well in our application. So we trained the YOLO in our own database (which is collected from the video of experiments) for the fishery application in the specific environment.

Our fishery monitoring system uses a similar method to train the YOLO detection system, where the system models the detection task as a regression problem. The system divides the image into 7×7 grid and uses each grid cell to predict two bounding boxes with confidence ($P_r(Object) * IOU_{pred}^{truth}$), where IOU is the intersection over union[8].

We select the threshold (0.5-0.7) to choose the suitable bounding box and avoid multi-object in the cell. For each bounding box, the results are (x, y, w, h) and confidence (five prediction value). Coordinate (x, y) represents the bounding box's center relative to the grid cell. (w, h) represents the width and height relative to the whole input image. Our system only has one labeled class (fish or not), so the result is encoded into a $7 \times 7 \times (2 * 5 + 1)$ tensor. The unified architecture and the training process of our detection network are the same as the network in the paper[8].

With images in the bounding box, we first do the binarization process to remove some unrelated pixels (like grass). And then we extract the edges of fishes with Canny operator. In this step, we keep the edges with the most continuous pixels as the fishes. So we get the pixels of the fishes to prepare for the computation of the size of fishes later.

2.4 Depth prediction with SGBM

Given the calibration parameters, we use the SGBM (Semi-global block matching) algorithm to predict the depth information. SGBM is a semi-global matching algorithm to compute the disparity for the binocular camera. With the calibration parameters and disparity image, we can predict the depth image by the principle of the stereo camera. (Figure 4)

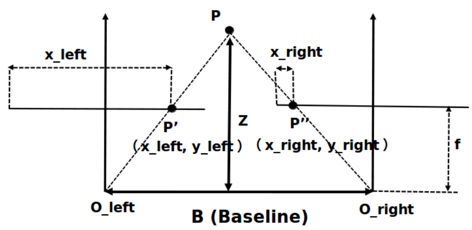

Fig. 4: Depth prediction with stereo camera

The first step is to complete the distortion correction and stereo correction, where the intrinsic and extrinsic parameters are needed. We can use Bouguet algorithm to reproject the left and right images so that they can be in the same plane where the rows of the image are aligned to the parallel structure. (Both correction process has its open access API, and many stereo cameras have already been aligned into the same plane directly.)

If the distortion is corrected and the image plane of the left/right cameras has been strictly aligned. From the stereo camera model (Figure 4), we can know that:

$$Z = \frac{f * B}{(x_{left} - x_{right})} \tag{9}$$

Where f is the focus, Z is the depth, P is the point in the real world. O_{left} and O_{right} are optical centers of the left and right image planes and P' and P" are the imaging point on the left and right image planes with the coordinate (x_{left}, y_{left}), (x_{right}, y_{right}). The value of $(x_{left} - x_{right})$ is the disparity of the two images, which is computed by the SGBM later.

The algorithm SGBM is to minimize an energy function of the disparity map to compute the disparity of each pixel[9]. The energy function is $E(D)$:

$$E(D) = \sum_p (C(p, D_p) + \sum_{q \in N_p} P_1 I[|D_p - D_q| = 1] \\ + \sum_{q \in N_p} P_2 I[|D_p - D_q| > 1]) \tag{10}$$

D is the disparity map, p and q are the pixels in the image. N_p is the adjacent pixel points of p (usually 8 points). $C(p, D_p)$ is the cost of the pixel p when its disparity is D_p. P_1 is the penalty values used for the adjacent pixels which satisfy the condition $|D_p - D_q| = 1$. P_2 is the penalty values used for the adjacent pixels which satisfy the condition $|D_p - D_q| > 1$. $I[.]$returns 1 when the parameter in the function is true, or 0 if not.

This optimal computation is a NP-complete problem, which costs too much. In practice, this problem is approximately decomposed into several one-dimensional problems. Each one-dimensional problem can be seen as a linear problem, which can be solved by dynamic programming. This solution is packaged in the OpenCV library, which we can directly use. Once we get the disparity map from the SGBM algorithm, then we can compute the depth maps of the fish we extracted from the bounding boxes.

2.5 The Fishery monitoring system

After the detection work and the computation of the depth map, we can estimate the size of fish directly by the equation (11), which is derived from Figure 5.

For one fish: $D \in (1, 2, 3, 4, 5)$

$$\begin{aligned} \underset{i \subseteq D}{L_i} &= \frac{l(Z-f)}{f} \\ L &= \underset{i \subseteq D}{max}(L_i) \end{aligned} \tag{11}$$

Where L is the size of fish in the real world, and l is the size of fish in the image $l = (u_1 - u_0)$, which can be represented by the coordinate value. Z is the depth of the plane (estimation) and f is the focus.

Fishes are nonrigid, so we use real-time computation and choose the most regular and biggest result to estimate the size of the fishes.

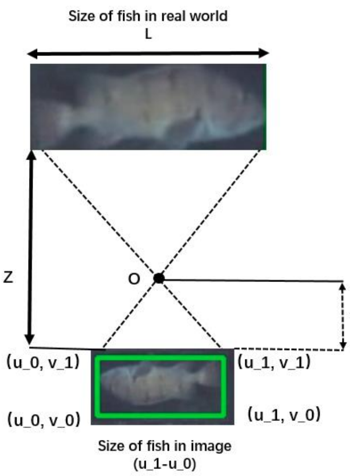

Fig. 5: The size estimation

3 Experiment

We do the experiment with this fishery monitoring system on the platform QLAUV-II. We also test it in some rough environment to check its robust performance.

3.1 Calibration

For the underwater application, we need to calibrate the stereo camera underwater by the method in section 2.2.
The calibration summary is:

1. Print a calibration board and put it into the water.
2. Keep the stereo camera still and take a few images of calibration board from different orientations.
3. Detect the feature points of the calibration board.
4. Estimate the intrinsic and extrinsic parameters
5. Refine parameters with the maximum-likelihood.

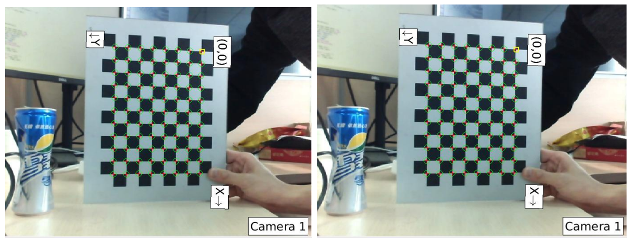

Fig. 6: The example of the calibration process

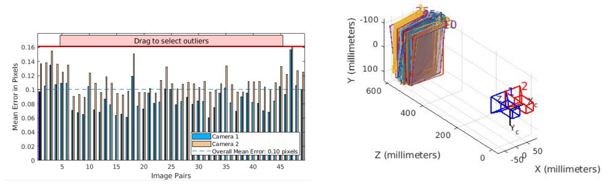

Fig. 7: The calibration results

The extrinsic matrix (rotation matrices and translation vector) can be computed by each picture. And from the tools, we can also compute the relative R and T matrix between two cameras:

$$R = \begin{bmatrix} 1.00 & -0.0004 & -0.003 \\ 0.0004 & 1.00 & 0.002 \\ 0.0026 & -0.002 & 1.00 \end{bmatrix}, T = \begin{bmatrix} -60 \\ 0.04 \\ 0.19 \end{bmatrix} \tag{12}$$

For the distortion vetor of two camera D_L (left) and D_R (right):

$$D_L = \begin{bmatrix} 0.06906 \\ -0.2746 \\ 0.00112 \\ -0.0003 \\ 0.43551 \end{bmatrix}, D_R = \begin{bmatrix} 0.06929 \\ -0.19845 \\ 0.000655 \\ -0.00044 \\ 0.147510 \end{bmatrix} \tag{13}$$

For the intrinsic matrix of left camera:

$$M_1 = \begin{bmatrix} 754.7735 & 0 & 337.1317 \\ 0 & 754.6285 & 242.2131 \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

For the intrinsic matrix of right camera:

$$M_1 = \begin{bmatrix} 763.0099 & 0 & 349.4117 \\ 0 & 763.1437 & 237.2877 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

We can see that from the experiment, the main calibration process has been complicated successful.

3.2 YOLO based fish detection system

We use the underwater video taken by AUV and calibrated camera parameters to test the performance of this monitoring system. To check its accuracy, we compare the Figure 8 and Figure 9. The good performance can be seen in Figure 9, where most of target fishes are detected.

Fig. 8: The test video

Fig. 9: The fish-detected video

3.3 Depth prediction and size estimation

To test this algorithm, we firstly complete the distortion correction and stereo correction. We use Bouguet algorithm to reproject the left and right images so that they can be in the same plane where the rows of the image are aligned to the parallel structure. The result then can be the input for the stereo matching like Figure 10.

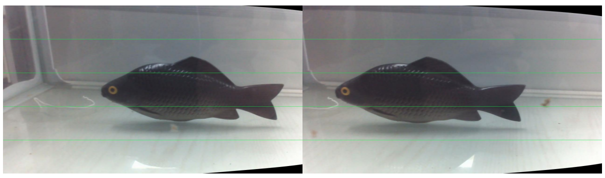

Fig. 10: The image after correction

With the bounding box from the corrected images, we extract the fishes by binarization process and Canny operator and we keep the edges with the most continuous pixels as the fishes in Figure 11.

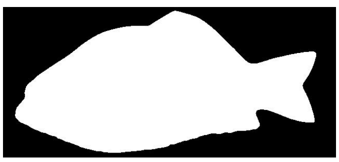

Fig. 11: The edge of fish from bounding box

With the input images and the parameters of stereo camera, we can compute the disparity and depth map with the method in section 2. In the distance range (25cm-75cm), Figure 12 shows the results of the SGBM. It shows that even in this rough environment, this algorithm is also very robust in depth prediction, which prepares for our system to monitor fishes in different environment later.

Fig. 12: The disparity map

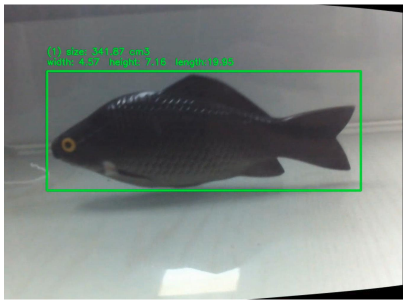

Fig. 13: The result of fish detection system

With the depth map of the fishes, we can get the real distance between any two pixels. So we can get length and the height of the fish. From the depth information, we can get the width information directly. So we get the size of the fishes. And the results show that the detection is real-time and accurate with a suitable edge from bounding box for each fish.

3.4 Accuracy

To compare the performance of our system. We monitor several fishes in different depth range for that the depth range affects the accuracy of our system. We compute the error rate compared with the true length. In the experiment, we compute the same fish 100 times in each depth range, and calculate the average error rate for each depth range in Table 1. We choose the depth range as (25cm-50cm), (50cm-75cm), (75cm-100cm). For that camera cannot get the full view of fish in the 0 cm-25 cm. And when depth value is bigger 100 cm, the algorithm is not practical with big errors.

We also compare our system with the template matching algorithm (TM) to show our advantages in Table 1. Template match method is another method to calculate the sizes of fishes in our system. The only difference is to change the SGBM algorithm to TM algorithm, where TM method can be use directly from the OpenCV as the 'cv2.matchTemplate' function. And from the Table 1 and experiments, we can see that TM makes faster fps but bring less accuracy. The our fishery system is also real-time fps and robust for its accuracy. Most of all, the SGBM based system can also get some width or height information to reconstruct the fishes in some future work.

Table 1: Performance of the system

	Average error rate of SGBM in 100 times	Average error rate of TM in 100 times
25 cm-35 cm	4.81%	7.66%
35 cm-50 cm	6.36%	8.32%
50 cm-75 cm	12.92%	16.76%
75 cm-100 cm	16.21%	21.21%

4 Conclusion

In this paper, we propose a new fishery monitoring system, which consists of AUV platform, binocular calibration, fish detection, depth prediction, and size estimation.

The whole system aims to monitor fishes by estimating the sizes of fishes with the visual method instead of the high cost and information-limited sonar measurement or manual method. Compared with TM method, our method is more accurate. And the experimental results indicate that in some depth range (25-50 cm), our new fishery monitoring system is an effective and efficient method for that we can control the error in 0.2-1.2 cm.

5 Future work

For that fishes are all nonrigid moving when we detecting, we can try some segmentation method for more accurate results instead of edges extraction. If we can get the whole segmentation, we can reconstruct the fish in 3-D in an ideal situation, which means not only the size but also the entire situation of fishes can be collected.

We are also testing the new system: Fishery Monitoring II for that the SGBM can reconstruct the width and heighth information like Figure 13. by fitting an ellipsoid with a neural network.

For that the system is operated underwater, the calibration process is necessary but a bit troubled. We can design some estimation method for the dry calibration like to avoid the effects of the reflection from water [11]. This calibration process may be combined with some state-of-art algorithm for the image process underwater like DCP[12] or some underwater image enhancement method[13].

References

[1] L. Kou and J. Xiang and J. Bian, Controllability Analysis of a Quadrotor-like Autonomous Underwater Vehicle, in *2018 IEEE 27th International Symposium on Industrial Electronics (ISIE)*, 2018.

[2] J. W. Kaeli, Real-time anomaly detection in side-scan sonar imagery for adaptive AUV missions, in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, 2016.

[3] Moursund, Russell A and Carlson, Thomas J and Peters, Rock D, A fisheries application of a dual-frequency identification sonar acoustic camera, in *ICES Journal of Marine Science*, vol.60, 2003.

[4] G. L. Foresti et al., A voting-based approach for fast object recognition in underwater acoustic images, in *IEEE Journal of Oceanic Engineering*, vol.22, 1997.

[5] M. F. Yahya and M. R. Arshad, Position-based visual servoing for underwater docking of an autonomous underwater vehicle, in *2016 IEEE International Conference on Underwater System Technology: Theory and Applications (USYS)*, 2016.

[6] G. L. Foresti, Position-based visual servoing for underwater docking of an autonomous underwater vehicle, in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol.31, 2001.

[7] Zhengyou Zhang, A flexible new technique for camera calibration, in *IEEE Transactions on pattern analysis and machine intelligence*, vol.22, 2000.

[8] Redmon, Joseph and Divvala, Santosh and Girshick, Ross and Farhadi, Ali, You only look once: Unified, real-time object detection, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[9] H. Hirschmuller, Stereo Processing by Semiglobal Matching and Mutual Information, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.30, 2008.

[10] Kou, Liwei and Xiang, Ji and Li, Yanjun and Bian, Jingwei, Stability and nonlinear controllability analysis of a quadrotor-like autonomous underwater vehicle considering variety of cases, in *International Journal of Advanced Robotic Systems*, vol.15, 2018.

[11] Huang, Longxiang and Zhao, Xu and Huang, Xingling and Liu, Yuncai, Underwater camera model and its use in calibration, in *2015 IEEE International Conference on Information and Automation*, 2015.

[12] He, Kaiming and Sun, Jian and Tang, Xiaoou, Single image haze removal using dark channel prior, in *IEEE transactions on pattern analysis and machine intelligence*, vol.33, 2011.

[13] Khan, Amjad and Ali, Syed Saad Azhar and Malik, Aamir Saeed and Anwer, Atif and Meriaudeau, Fabrice, Underwater image enhancement by wavelet based fusion, in *2016 IEEE International Conference on Underwater System Technology: Theory and Applications (USYS)*, 2016.