International Journal of Innovative Computing, Information and Control
Volume **21**, Number **5**, October **2025**

ICIC International ©2025 ISSN 1349-4198
pp. **1383–1399**

A ROBUST 3D POSITION ESTIMATION AND TRACKING METHOD OF MULTIPLE FISH FOR OCCLUSION AND MIRROR IMAGE IN A SMALL TANK

Hiroki Yamaguchi[1,*], Katsumi Tateno[1,2] and Keiichi Horio[1]

[1]Graduate School of Life Science and Systems Engineering
[2]Research Center for Neuromorphic AI Hardware
Kyushu Institute of Technology
2-4 Hibikino, Wakamatsu-ku, Kitakyushu-shi, Fukuoka 808-0196, Japan
tateno.katsumi220@mail.kyutech.jp; horio@brain.kyutech.ac.jp
*Corresponding author: yamaguchi.hiroki246@mail.kyutech.jp

Received December 2024; revised April 2025

Abstract. *Animal populations exhibit adaptive changes in response to stimuli. A time series of individual positions and poses must be tracked to reveal collective animal behavior. Tracking fish in 3D space is needed, especially for fish. However, tracking in camera images involves the difficulties of occlusions and mirror images. This study uses three cameras to propose a robust 3D position estimation and tracking method of multiple fish for occlusion and mirror image in a small tank. Fish occlusion occurring in one direction is handled by estimating the 3D position using results from the other two directions. DeepLabCut for multiple animals (maDLC) finds fish feature points for each camera view. Post-processing performs for proper correspondence between feature points and individuals: Removing misassignments, matching the feature point in all camera views, and merging feature points representing the same fish. Feature points are estimated in 3D, and if their centroid is outside of the tank, they are eliminated as mirror images. Experimental results showed that even if fish occlusion occurred in one direction, 3D position estimation could be performed if feature points were estimated in the other two directions using the proposed method. Mirror images were eliminated, and the actual fish could be tracked.*

Keywords: Collective behavior, Multiple fish, 3D position estimation, Tracking, Occlusion, Mirror image

1. **Introduction.** Fish behave collectively depending on their species; however, it is still not fully clarified why they behave collectively, how they do so, what evolutionary processes led to the acquisition of collective behavior, and how collective behavior developed during growth [1]. Tracking a time series of individual positions and poses is needed to reveal collective fish behavior. Computerized information engineering approaches have made it possible to track them automatically. However, the fish to be tracked may hide behind other fish (occlusion), making tracking difficult in the collective behavior of fish in a tank. Furthermore, the mirror images that occur when fish approach the water surface, sides, or bottom of the tank obscure the positions of the actual fish. Several studies have detected and tracked multiple fish in 3D, considering occlusion and mirror-image problems. Multiple fish 3D detection and tracking methods using two cameras have been proposed [2,3]; however, if occlusion occurs in one direction, fish may not be detected and tracked. Therefore, if considering occlusion, a minimum of three cameras is required. This study uses three cameras to propose a robust 3D position estimation and tracking method

DOI: 10.24507/ijicic.21.05.1383

for multiple fish in a small tank. Fish occlusion occurring in one direction is handled by estimating the 3D position using results from the other two directions, and if the fish's estimated 3D position is positioned outside the tank, it is eliminated as a mirror image. The proposed method has the following advantages: 1) The 3D position and pose of each fish can be obtained in a time series; 2) Compared to the previously proposed two-camera method [2], the three-camera method of this study can estimate more 3D positions; 3) Even if fish occlusion occurs in one direction, 3D position estimation can be performed if feature points are estimated in the other two directions; 4) It eliminates mirror images and tracks actual fish.

2. **Related Work.** Numerous studies have detected and tracked fish, and several studies have detected and tracked multiple fish in 3D, considering occlusion and mirror-image problems. Verschae et al. [4] set up three cameras in different directions outside a small tank to perform 3D detection and tracking of multiple fish in real time by triangulation and silhouette matching using a varifocal camera model [5] that considers light refraction. The method is robust to occlusion. However, that study mainly evaluated processing time, not detailed evaluations of 3D detection and tracking, occlusion, and identifying whether the fish were mirror images or not. Therefore, detailed evaluations of them are necessary. Palconit et al. [3] set up two web cameras on the top of a small tank and performed multiple fish 3D tracking using binarization for fish detection, the k-nearest neighbor (KNN) algorithm for fish matching, triangulation for depth calculation, and various tracking algorithms. The study used cameras only on the top; therefore, if occlusion occurs in one camera, fish may not be detected and tracked. It is more robust against occlusion to use multiple cameras to take images from multiple directions. Furthermore, the study did not mention mirror images. The authors [2] placed cameras on the top and front of a small tank, transformed the fish feature points estimated from the two camera views and the camera points to the world coordinate system, and estimated the 3D positions of multiple fish by calculating the shortest distance between the lines passing through the midpoint of the feature points and the camera point. Although the method is robust to mirror images, if the fish feature points cannot be estimated due to occlusion in one camera view, the 3D position cannot be estimated either. By improving the method by increasing the number of cameras in different directions from two to three, it is expected to increase the number of 3D position estimates and be more robust against occlusion. This study uses three cameras to perform robust 3D position estimation and multiple fish tracking for occlusion and mirror image in a small tank. Furthermore, we evaluate in detail the accuracy of 3D position estimation and tracking, and the robustness of occlusion and mirror image.

Recently, tools have been developed to detect and track target animals [6-13]. Several software programs are capable of detecting 3D postures or multiple animals. DeepLabCut (DLC) is an open-source software for animal behavior analysis [10,11]. DLC is based on DeeperCut [14], a bottom-up human pose estimation method. DeeperCut uses a Residual Network (ResNet) [15] and can accurately estimate human body parts from images. DLC adds deconvolution layers as an output to the ResNet structure of DeeperCut. The deconvolution layers represent each body part position as a different image and learn the confidence level of each image. ResNet is pre-trained on ImageNet [16]. This transfer learning enables feature point estimation and tracking of the target animal by learning additional label data for a small number of video frames provided by the user. DLC is available for multiple animals (maDLC) [12]. The maDLC can use a multi-scale architecture (DLCRNet_ms5) that adapts Resnet and EfficientNet [17,18], which are pre-trained

on ImageNet. This architecture can be learned to estimate body part positions, connections, and individuals. However, some problems have been reported, such as the inability to track the estimated feature points successfully. Furthermore, maDLC directly supports 3D pose estimation using multiple cameras; however, it is currently limited to two cameras. There is an open-source toolkit called Anipose [13] that can utilize DLC to estimate the 3D pose of an animal from two or more camera views; however, it is not available for multiple animals. In this study, we used maDLC only for fish feature point estimation in camera images.

3. **Method.** This chapter describes the proposed method and experimental setup. Figure 1 shows the overall flow of the proposed method.

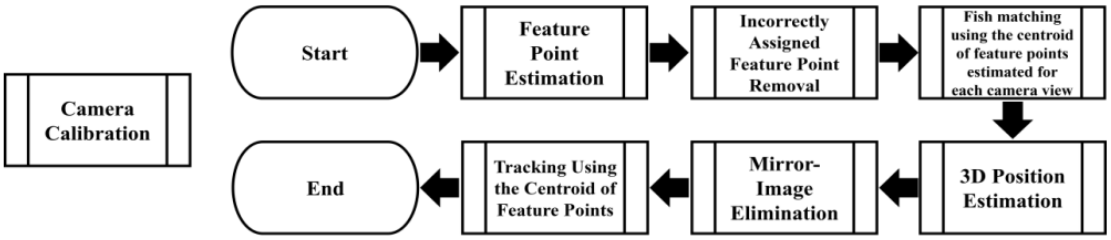


FIGURE 1. Overall flow of the proposed method

3.1. **Camera calibration.** Camera calibration is to estimate the parameters that represent the characteristics, position and orientation of the camera used from the taken images. This study dealt with geometric calibration, which is a type of camera calibration that estimates internal parameters representing camera characteristics such as focal lengths, image center, and lens distortion coefficients, and external parameters representing camera position and pose. The estimated parameters were used to correct for image distortion. Furthermore, they were used for the transformation of the target point from the image coordinate system to the world coordinate system (used in Section 3.3). Figure 2 shows the overall flow of camera calibration.

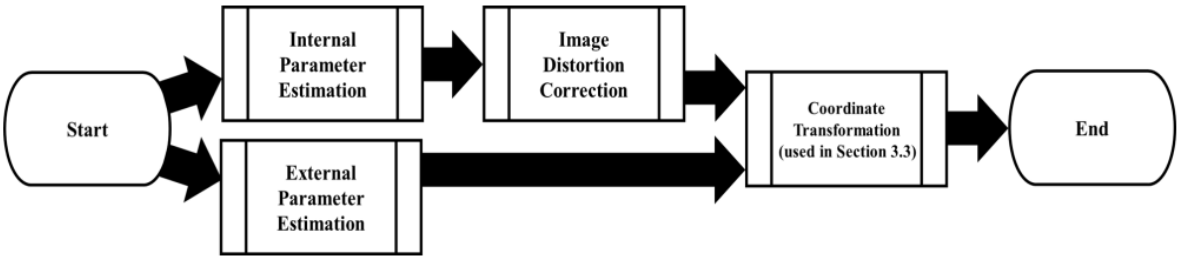


FIGURE 2. Overall flow of camera calibration

Camera calibration requires the determination of a camera model. We defined a commonly used pinhole camera model. Zhang's method was used to estimate internal parameters [19]. The external parameters were estimated from the correspondence between the positions of multiple points in the image coordinate system and the positions of multiple points in the world coordinate system of the object taken from the camera positions set up in our experiment. Internal and external parameters can be estimated using the Open Source Computer Vision Library (OpenCV) [20].

Actual camera lenses have radial and circumferential distortion. When a point in 3D space is taken and the image coordinates are obtained, if the lens is distorted, the coordinates of the point can be represented by the following equations.

$$x_d = x_u\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + 2p_1 x_u y_u + p_2\left(r^2 + 2x_u^2\right) \tag{1}$$

$$y_d = y_u\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + p_1\left(r^2 + 2y_u^2\right) + 2p_2 x_u y_u \tag{2}$$

$$r^2 = x_u^2 + y_u^2 \tag{3}$$

(x_u, y_u) are the coordinates with an ideal lens without distortion, (x_d, y_d) are the coordinates with a lens with distortion, and r is the distance from the image center to (x_u, y_u). k_i $(i = 1, 2, 3)$ are the radial distortion coefficients and p_j $(j = 1, 2)$ are the circumferential distortion coefficients.

The coordinate transformation from the image coordinate system to the world coordinate system of the target point by the pinhole camera model is represented by the following equation:

$$M = R^{-1}\left(A^{-1}\lambda \boldsymbol{m} - \boldsymbol{t}\right)$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}^{-1} \left(\begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \right) \tag{4}$$

This equation transforms the target point into the image coordinate system, the camera coordinate system, and the world coordinate system, in that order. (u, v) represents 2D coordinates in the image coordinate system, $\boldsymbol{m} = [u, v, 1]^T$ represents its homogeneous coordinates. $M = [X, Y, Z]^T$ represents a 3D point in the world coordinate system. λ is the scale coefficient of the image, which represents the instability that depth cannot be obtained from a single image alone when considering the projection of an image from a 2D point to a 3D space. A is the internal parameter matrix of the camera and consists of the focal length f_x, f_y expressed in pixels, the image center (u_0, v_0), and the skew coefficient γ representing the orthogonality of the image coordinate axes (treated as $\gamma = 0$ in OpenCV functions). The external parameters consist of a rotation matrix R and a translation vector $\boldsymbol{t}$. They transform the target point from the camera coordinate system to the point M in the world coordinate system. This transformation is equivalent to the case $z \neq 0$ in the following Equations (5)-(9) [21-24].

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R^{-1}\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} - \boldsymbol{t} \right) \tag{5}$$

$$x' = x/z \tag{6}$$

$$y' = y/z \tag{7}$$

$$u = f_x * x' + u_0 \tag{8}$$

$$v = f_y * y' + v_0 \tag{9}$$

3.2. **Feature point estimation and incorrectly assigned feature point removal.** This section describes feature point estimation of fish body parts using maDLC and incorrectly assigned feature point removal. The results of feature point estimation are saved in a CSV file. The CSV file contains the coordinates of the feature points estimated from the images and the likelihood $(0 \sim 1)$, indicating the validity of the estimation for each individual. Even when this likelihood is close to 1, the feature points of one individual are sometimes estimated to belong to others. Therefore, we performed the following procedures to remove incorrectly assigned feature points.

Step 1) Fish regions were extracted by the frame subtraction method. The threshold for binarization was automatically obtained using the discriminant analysis method

(sometimes called the Otsu method [25]). The masked image in which the background and the moving object regions are separated contains small holes. If the holes are found in the fish region, they affect the incorrectly assigned feature point removal. Therefore, we removed these by closing, which performs the same number of dilations and erosions [24].

Step 2) Figure 3 shows how to remove incorrectly assigned feature points. First, we draw a rectangle with the snout tip and tail tip feature points as its diagonal. Second, we count the number of pixels on a diagonal line connecting the two feature points. We then count the number of pixels where the diagonal line overlaps with the fish region in the rectangle. Finally, we calculate the ratio of those pixel counts. We named it the fish-over-line rate. If the value of the fish-over-line rate is low, two feature points of different fish are likely connected. Appropriate thresholding can remove incorrectly assigned feature points.

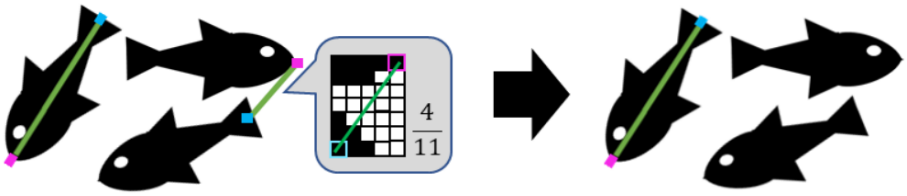


FIGURE 3. (color online) Incorrectly assigned feature point removal. If the snout tip (purple square) and tail tip (blue square) are feature points of the same individual, the diagonal line between the feature points almost overlaps with the fish region. The fish-over-line rate is low if the two feature points are from different fish. Such incorrectly assigned feature points should be removed.

3.3. **Fish matching using the centroid of feature points estimated for each camera view and 3D position estimation.** Figure 4 shows the overall flow of fish matching using the centroid of feature points estimated for each camera view and 3D

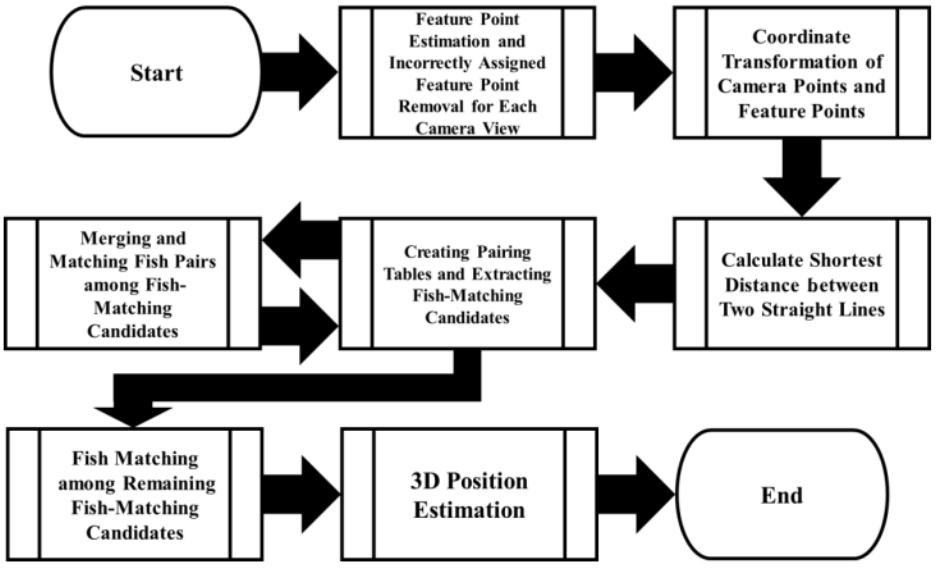


FIGURE 4. Overall flow of fish matching using the centroid of feature points estimated for each camera view and 3D position estimation

position estimation. Fish matching using the centroid of feature points estimated for each camera view and 3D position estimation were performed as follows.

Step 1) Fish feature point estimation and incorrectly assigned feature point removal from videos taken by three cameras (snout tip, center, and tail tip as feature points).

Step 2) Coordinate transform camera points and multiple fish feature points. Camera points are transformed from the camera coordinate system to the world coordinate system, and multiple fish feature points are transformed from the image coordinate system to the world coordinate system (once transformed to 0 depth coordinates because the depth is unknown).

Step 3) Draw straight lines connecting the camera point and the centroid of three feature points, and calculate the shortest distance between two straight lines in every pair of camera views (Top & Front, Front & Side and Side & Top).

Figure 5 shows a diagram illustrating the process of finding a set of centroids representing an identical position from the three camera views and matching the fish. In searching for an appropriate set of centroids, we use the previous study's findings that the distance at which neighboring fish cannot get any closer is about 30% of their body length [26]. If the distance between the centroids is shorter than the nearest-neighbor distance, they should belong to the same fish. However, since it is not certain that the centroids can be found in all camera views, the centroids can be merged and associated with fish in the following steps.

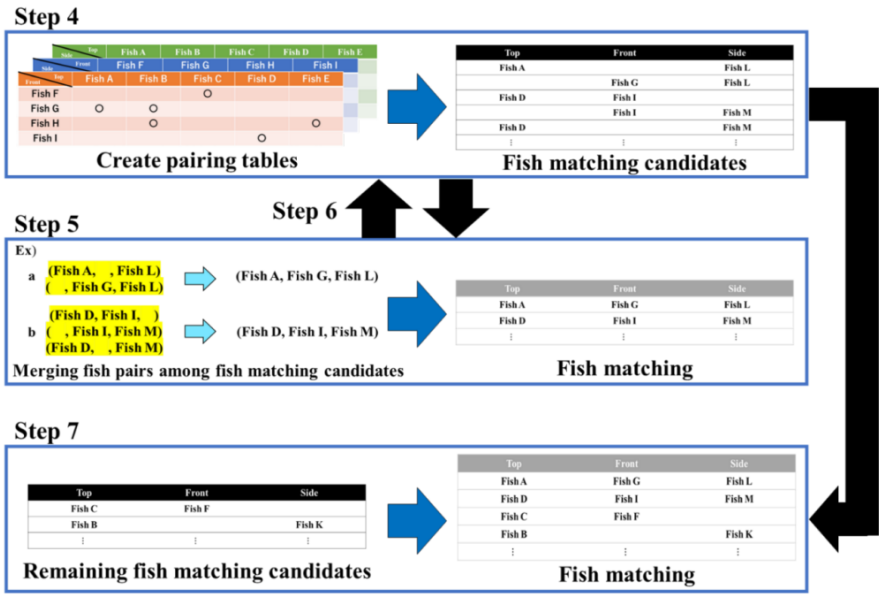


FIGURE 5. Fish matching using the centroid of feature points estimated for each camera view. In Step 4, fish-matching candidates are extracted from the pairing table. In Step 5, among the fish-matching candidates, if the fish pairs can be merged, they are merged, and the fish are matched. Furthermore, from the three pairing tables and fish-matching candidates in Step 4, remove those that overlap with the matched fish. In Step 6, return to Step 4, and if no new fish-matching candidates can be extracted from the three pairing tables in Step 4, proceed to Step 7. In Step 7, among the remaining fish-matching candidates, if a fish pair does not overlap with any other fish pair and does not overlap with a fish that has already been matched, the fish pair is matched.

Step 4) First, we assign a tentative label to the centroids in each camera view (e.g., Fish A). These labels are made differently across all camera views. For example, we might label Fish A through E in the top camera view, but in the side camera view, we start with Fish F. Then, we prepare a pairing table for each pair of camera views (Figure 5 (Left-Above)). If centroids are found in the camera view pair whose inter-centroid distance is less than 30% of the body length, they are marked (with a circle) in the cell of the fish pair corresponding to the respective centroid in the pairing table. If a marked fish pair does not overlap with any other marked fish pair in the pairing table, the feature points of that fish pair are extracted as a fish matching candidate (Figure 5 (Left-Low)). If no new fish-matching candidates can be extracted from the three pairing tables, proceed to Step 7.

Step 5) Among the fish-matching candidates, if a fish pair overlaps with another fish pair, the cases are divided as follows.

a) If there is only one fish pair that overlaps with another fish pair with each other, the two fish pairs are merged and the fish are matched (Figure 5 (Middle-Above a, Middle-Low)). However, the blank spaces must not overlap.

b) If there are only two fish pairs that overlap with a fish pair and those two fish pairs overlap with each other, the three fish pairs are merged and the fish are matched (Figure 5 (Middle-Above b, Middle-Low)). However, the blank spaces must not overlap.

After matching fish, remove those that overlap with the matched fish from the three pairing tables and fish-matching candidates in Step 4.

Step 6) Return to Step 4.

Step 7) Among the remaining fish-matching candidates, if a fish pair does not overlap with any other fish pair and does not overlap with a fish that has already been matched, the fish pair is matched (Figure 5 (Right-Above and Low)).

Step 8) Figure 6 shows a schematic picture of the 3D position estimation. Draw straight lines (straight lines n, o and p in Figure 6) connecting the camera point and one of the matched fish's three feature points, and obtain the coordinates of the points

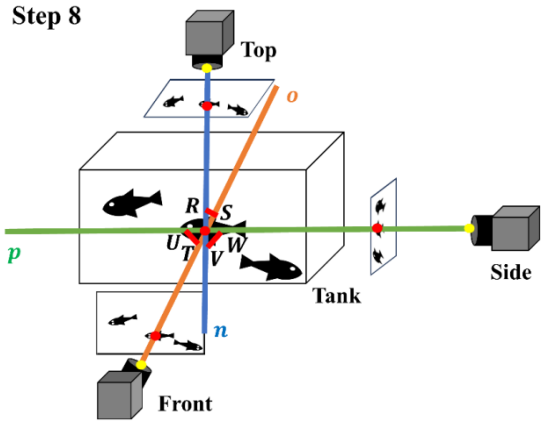


FIGURE 6. 3D position estimation. After transforming the camera point and the multiple fish feature points estimated for each camera view into 3D points in the world coordinate system, fish matching and 3D position estimation are performed using the shortest distance calculation between two straight lines passing through the camera point and centroid of the feature points. Mirror images are also performed without distinction.

R, S, T, U, V and W on those line segments by calculating the shortest distance between the two straight lines in two directions. The 3D position is taken as the centroid of the coordinates of the bisecting points of the line segments RS, TU, and VW. If only two bisecting points can be obtained, the mean of those two points is the 3D position.

3.4. **Mirror-image elimination and tracking using the centroid of feature points.** The mirror-image elimination and tracking using the centroid of feature points were performed as follows.

Step 1) Fish's 3D position estimation and mirror-image elimination in every frame.
Step 2) Assign tracking IDs to the centroids of all individual feature points in the first frame.
Step 3) The fish with the shortest Euclidean distance between the centroids in the two sequential frames are considered to be the same individual and are tracked. The maximum distance is the mean length of the fish.
- If no applicable point exists, a new tracking ID is assigned, not exceeding the number of fish. If no assignment is necessary, estimation is performed between the next frames.
- If there is an applicable point, linear interpolation is performed between the two frames.

For Step 1, the centroid of the three estimated fish 3D feature points was used, and if it was positioned outside the centroid of the 3D point at the corner of each face of the tank, which was estimated using images taken of the tank, it was identified as a mirror image and eliminated.

3.5. **Experimental setup.** All the experimental protocols were reviewed by the Animal Institutional Review Board of the Kyushu Institute of Technology and approved by the president of Kyushu Institute of Technology. We used a PC with the following specifications for our experiments.

PC: THIRDWAVE Diginnos PC
OS: Windows 10 Education
CPU: Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
RAM: 8.00GB
GPU: NVIDIA GeForce GTX 1070
VRAM: 8.00GB

We created and ran the programs using Jupyter Notebook (Python 3.9.13), OpenCV 4.7.0 functions for camera calibration, and DeepLabCut 2.3.0 for feature point estimation.

We set up a Sony action camera (HDR-AS50) at three positions from the top, front, and side of the tank (width: 390 mm, height: 200 mm, depth: 265 mm, thickness: 5 mm, water depth: 150 mm, water temperature: 25.5°C, pH: 7.7) to synchronously take several videos of fish or plastic fish models in the tank for use in our experiments. The distance from the tank to the camera lens was approximately 300 mm from the top, 220 mm from the front, and 290 mm from the side. We set the camera's video resolution to 1080p (1920×1080 pixels) and the frame rate to 30 fps.

To estimate the internal parameters of the camera, we prepared images of the chessboard pattern taken at various positions and orientations. The chessboard pattern has a length of 33 mm per square and a total number of inner corners of 6×8. The internal parameters were estimated from 23 images in which the corners of the chessboard pattern were detected using OpenCV, and the distortion of the images used in our experiments was corrected. When we corrected the image distortion, unwanted black pixels appeared

in the corners of the images; thus, we cropped them out, resulting in images of 1710×785 pixels in size. To estimate the external parameters of the camera, we used the coordinates of the corners of the tank.

4. **Experimental Results and Discussion.** This chapter describes the contents, results, and discussion of our experiments.

4.1. **3D position estimation with plastic fish models.** We placed three plastic fish models (Figure 7) in the tank. Two of them were oriented horizontally, and one was oriented diagonally. Its coordinate (x, y, z) of the snout tip, center, and tail tips were obtained directly from the three patterns of images in Figure 8. Each coordinate axis is as shown in Pattern 1 of Figure 8. The circles in Figure 8 indicate mirror images. We

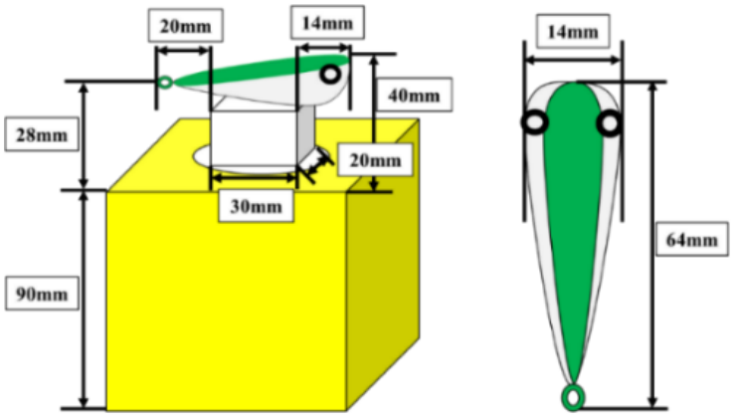
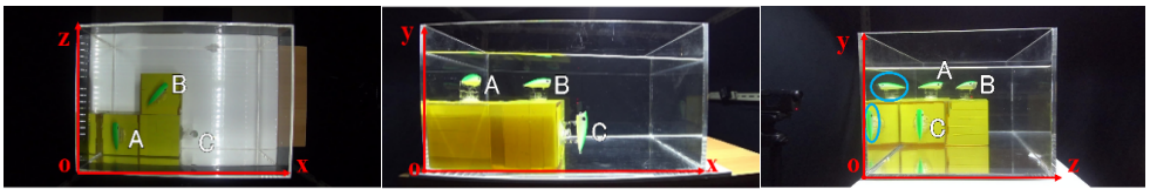
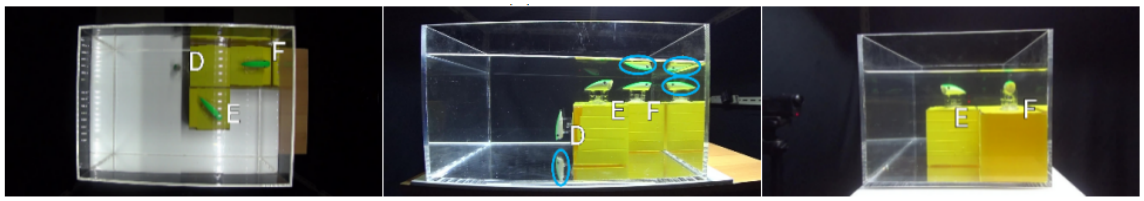


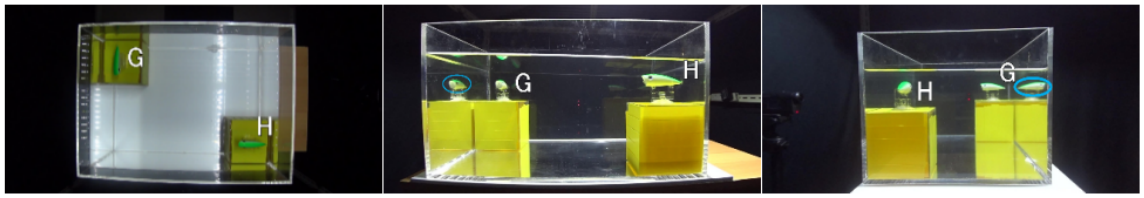
FIGURE 7. Plastic fish model



(a) Pattern 1



(b) Pattern 2



(c) Pattern 3

FIGURE 8. Plastic fish model positions (Left: top view, Center: front view, Right: side view)

estimated the 3D positions of these points using our previous method [2] and the proposed method and compared the number of estimates. In the previous method, we used top and front cameras. Table 1 shows the 3D position estimation comparison results.

TABLE 1. 3D position estimation of the plastic fish model. S is the snout tip, C is the center, and T is the tail tip.

Symbol		Actual coordinate [mm]	Estimated coordinate by our previous method [mm] [2]	Estimated coordinate by the proposed method [mm]
A	S	(50, 135, 85)	–	(50, 140, 78)
	C	(50, 129, 53)	–	(50, 133, 42)
	T	(50, 123, 21)	–	(51, 125, 6)
B	S	(157, 135, 157)	(159, 136, 165)	(159, 137, 167)
	C	(135, 129, 135)	(136, 129, 140)	(135, 129, 141)
	T	(112, 123, 112)	(112, 121, 115)	(112, 122, 114)
C	S	(225, 85, 50)	–	(228, 81, 41)
	C	(219, 53, 50)	–	(222, 47, 40)
	T	(213, 21, 50)	–	(218, 12, 39)
D	S	(165, 21, 215)	–	–
	C	(171, 53, 215)	–	–
	T	(177, 85, 215)	–	–
E	S	(267, 135, 108)	(275, 136, 103)	(275, 136, 101)
	C	(245, 129, 130)	(251, 128, 125)	(252, 129, 124)
	T	(222, 123, 153)	(226, 120, 147)	(226, 121, 148)
F	S	(375, 135, 215)	(382, 134, 210)	(382, 134, 210)
	C	(343, 129, 215)	(349, 129, 211)	(349, 129, 212)
	T	(311, 123, 215)	(316, 123, 212)	(316, 123, 212)
G	S	(50, 135, 186)	–	(47, 139, 185)
	C	(50, 129, 218)	–	(47, 132, 221)
	T	(50, 123, 250)	–	(46, 124, 257)
H	S	(311, 135, 50)	(312, 135, 48)	(312, 135, 48)
	C	(343, 129, 50)	(345, 128, 47)	(344, 129, 47)
	T	(375, 123, 50)	(377, 121, 46)	(377, 121, 46)

From the results in Table 1, the proposed method using three cameras increased the number of estimates compared to the previous method using two cameras. The fish model D could only obtain the position in one direction from the images; therefore, neither method could estimate the 3D positions.

Next, Figure 9 shows the placement of the plastic fish model when viewed from the top and the Euclidean distance between the actual and estimated coordinates. Since the proposed method does not take account of the light refraction, it was expected that the Euclidean distance values for the plastic fish models B and G, which are far from the camera, would be larger than those for the plastic fish models A, C, and E, etc. However, the Euclidean distance values for the plastic fish models A, C, and E, etc., were larger than those for the plastic fish models B and G. These results show that other effects are more pronounced than the effect of light refraction. Although the refractive index is higher in the case of low water temperatures or seawater, the effect of light refraction is expected to be small due to the small size of the tank. In the case of large fish, the tank to

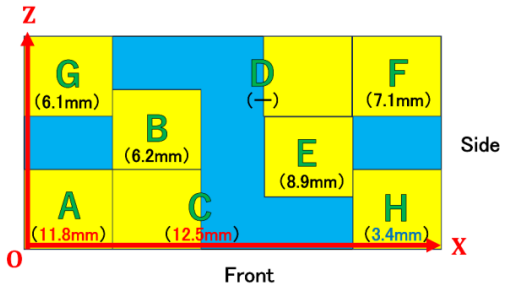


FIGURE 9. Placement of the plastic fish model when viewed from the top and the Euclidean distance between the actual and estimated coordinates

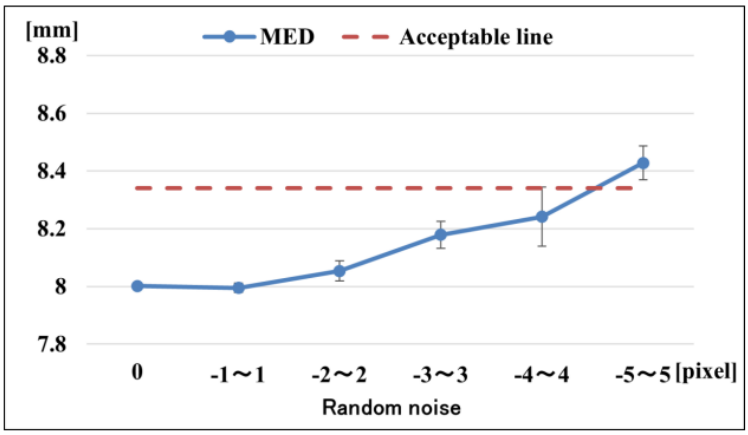


FIGURE 10. MED results in 3D position estimation using the proposed method when feature point positions, including random noise, are obtained directly from the images. The acceptable line represents 30% of the mean length of five Lambchop rasbora used in the subsequent tracking experiments.

be prepared is also large, which is expected to have different effects, such as water clarity, in addition to light refraction.

Furthermore, we obtained the mean Euclidean distance (MED) of 10 trials between the actual and estimated coordinates in 3D position estimation using the proposed method when the feature point positions, including random noise, are obtained from the images. Figure 10 shows the MED results in 3D position estimation using the proposed method when feature point positions, including random noise, are obtained directly from the images.

Three types of errors occur in 3D position estimation. Those errors are the estimation errors caused by obtaining the feature point positions from images, the errors caused by camera calibration when transforming the coordinates from the image coordinate system to the world coordinate system, and the errors caused by the effect of light refraction. Furthermore, fish have a nearest-neighbor distance that neighboring fish cannot get any closer to, about 30% of their body length [26]. If the MED between the actual and estimated coordinates is shorter than the nearest-neighbor distance, it is acceptable as an error. In Figure 10, the MED results when random noise is 0 pixels do not include the estimation errors that occur when obtaining the feature point positions from the images; otherwise, the results include three types of errors. In the subsequent tracking experiments, the acceptable line represents 30% of the mean length of five Lambchop rasbora (*Trigonostigma espei*). Error bars are standard errors. The results show that

errors in 3D position estimation are acceptable if the errors in obtaining the feature point positions are within $-3 \sim 3$ pixels.

4.2. **Feature point estimation for multi-fish in each camera view.** We used maDLC to estimate the positions of the fish's snout tip, center, and tail tips from images of the fish swimming in the tank, as shown in Figure 11. The maDLC outputs the feature point positions estimated from images for each individual; however, there is a possibility that the feature points of other individuals are included. In this case, point confusion with other individuals is possible when estimating later 3D positions. Therefore, in addition to evaluating feature point estimation for all individuals, we evaluated feature point estimation for each individual. We evaluated feature point estimation for all individuals using Precision and Recall in Equations (10) and (11) and for each individual using Precision′ and Recall′ in Equations (12) and (13). Furthermore, we calculated the mean Euclidean distance (MED) between the positions obtained directly from the images and the estimated positions in the feature point estimation for all individuals.

$$Precision = \frac{Number\ of\ feature\ points\ correctly\ estimated}{Number\ of\ estimated\ points} \tag{10}$$

$$Recall = \frac{Number\ of\ feature\ points\ correctly\ estimated}{Number\ of\ actual\ feature\ points} \tag{11}$$

$$Precision' = \frac{Number\ of\ feature\ point\ sets\ correctly\ estimated}{Number\ of\ estimated\ point\ sets} \tag{12}$$

$$Recall' = \frac{Number\ of\ feature\ point\ sets\ correctly\ estimated}{Number\ of\ actual\ feature\ point\ sets} \tag{13}$$

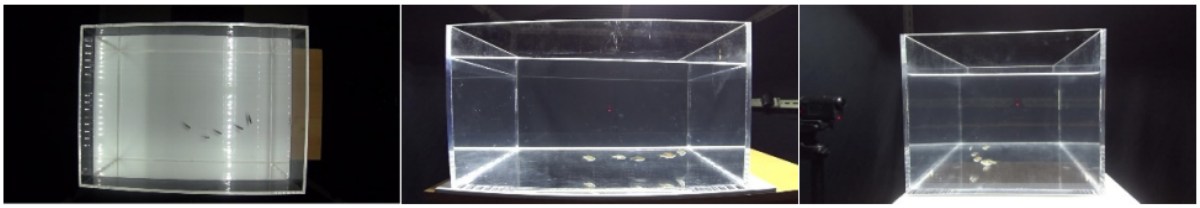


FIGURE 11. Fish swimming in the tank (Left: top view, Center: front view, Right: side view)

In this experiment, we used a group of five Lambchop rasbora of 278 ± 21 mm in body length. We took 300 frames in each direction from the videos and manually labeled the positions of the fish's snout tip, center, and tail tips. Mirror images were also labeled without distinction. These were used as training data and extended using the imgaug library [27], and the models were generated by training on top, front, and side views, respectively. DLCRNet_ms5 [12] was used as the network of maDLC. In the learning, we used an adaptive moment estimation (Adam) algorithm with set learning rates (0.0001 until 7,500 times, then 5×10^{-5} until 12,000 times, then 1×10^{-5} after that). The batch size was set to 8 by default, and the number of training iterations was set to 100,000. These were set to the recommended defaults. The threshold for the likelihood of feature point estimation was set to the default of 0.6, and the coordinates of feature points whose likelihood was smaller than the threshold were not used in this experiment. For the closing, kernels of size 5×5 with all values of 1 were used, and dilations and erosions were performed four times, respectively. We took 10 images in each direction from the videos not used in the training and used them as test data. Table 2 shows the feature point estimation results for all individuals. Precision was 1.000 in all three directions, and all estimated

TABLE 2. Feature point estimation results for all individuals

View	Image	Precision	Recall	MED [pixel]
Top	1st-10th	1.000	0.895	3.0
Front	1st-10th	1.000	0.842	2.2
Side	1st-10th	1.000	0.810	2.7

TABLE 3. Feature point estimation results for each individual

View	Image	Threshold	Precision′	Recall′
Top	1st-10th	0% (no removal)	0.981	0.883
		50%	0.972	0.583
		60%	0.960	0.400
		70%	0.955	0.350
		80%	0.941	0.178
		90%	1.000	0.133
		100%	1.000	0.033
Front	1st-10th	0% (no removal)	0.961	0.811
		50%	0.973	0.789
		60%	1.000	0.756
		70%	1.000	0.667
		80%	1.000	0.622
		90%	1.000	0.544
		100%	1.000	0.144
Side	1st-10th	0% (no removal)	0.967	0.797
		50%	0.960	0.649
		60%	1.000	0.621
		70%	1.000	0.568
		80%	1.000	0.459
		90%	1.000	0.338
		100%	1.000	0.095

feature points were correct. Recall was around 0.800 in all three directions, with feature points that could not be estimated. The MED between the positions obtained from the images and the estimated positions was less than 3 pixels. Table 3 shows the results of the feature point estimation for each fish. The estimated feature points are removed if the fish-over-line rate is lower than a threshold. The thresholds for the fish-over-line rate in the incorrectly assigned feature point removal were set as follows: 0% removal (no removal), 50%, 60%, 70%, 80%, 90%, and 100%. We calculated Precision′ and Recall′ to evaluate whether the snout tip, center, and tail tips of each fish could be estimated as a single set. Higher thresholds improved Precision′ whereas Recall′ worsened. That is, while the incorrect feature points were removed, the correct ones were also removed. It is expected that the misalignment of the estimated feature points and the bending of the fish's body when it changes direction had effects. If Precision′ is low, there is a possibility of point confusion with other individuals in the 3D position estimation. Even if Recall′ is a little low, its effect is small. This is because the 3D positions are estimated using points from three directions.

4.3. **Tracking multi-fish in a small tank.** We applied our proposed method to 30 consecutive frames (Data 1 and 2) of fish swimming in the tank from videos and tracked

them. We used the same five fish as in Section 4.2 as one group for this experiment. We obtained the fish's feature points directly every 5 frames, estimated their 3D positions, and calculated the mean nearest-neighbor distance and mean distance between centroids using the Euclidean distance. For Data 1, the mean nearest-neighbor distance = 55.4 mm and the mean distance = 163.3 mm. For Data 2, the mean nearest-neighbor distance = 32.1 mm, and the mean distance = 147.0 mm. Based on the experimental results in Section 4.2, the thresholds for the fish-over-line rate in the incorrectly assigned feature point removal were set as follows: 0% removal (no removal), 60%, and 90%. Precision″ and Recall″, which evaluate 3D position estimation, MOTA, which evaluates tracking, ID switch (IDSW), total number of IDs switched, and MAX, the maximum number of tracks simultaneously (more than 2 frames) were used as metrics. The equations for Precision″, Recall″, and MOTA are shown in Equations (14)-(16). fp_t, fn_t, $ID\ switch_t$, and g_t in Equation (16) are the number of incorrectly tracked, the number of untracked, the number of ID switched, and the number of fish to track in frame t. Tables 4 and 5 show the tracking results in Data 1 and Data 2, respectively.

$$Precision'' = \frac{Total\ number\ of\ fish\ correctly\ estimated\ in\ all\ frames}{Total\ number\ of\ estimates\ in\ all\ frames} \tag{14}$$

$$Recall'' = \frac{Total\ number\ of\ fish\ correctly\ estimated\ in\ all\ frames}{Total\ number\ of\ actual\ fish\ in\ all\ frames} \tag{15}$$

$$MOTA = 1 - \frac{\sum_t (fp_t + fn_t + ID\ switch_t)}{\sum_t g_t} \tag{16}$$

Table 4. Tracking results in Data 1

Threshold	Precision″	Recall″	MOTA	IDSW	MAX
0% (no removal)	1.000	0.658	0.707	0	4
60%	1.000	0.580	0.600	0	3
90%	0.976	0.267	0.460	0	3

Table 5. Tracking results in Data 2

Threshold	Precision″	Recall″	MOTA	IDSW	MAX
0% (no removal)	0.851	0.420	0.273	5	3
60%	0.780	0.213	0.167	2	2
90%	0.810	0.060	0.107	0	1

Our proposed method tracked up to four fish in Data 1 and three in Data 2 simultaneously. Continuous tracking and tracking of other fish were not possible due to the removal of even correctly assigned feature points by the fish-over-line rate threshold, failure to narrow down the target in fish matching, point confusion with other individuals in 3D position estimation, and switching of tracking targets. The distances between the fish in the group of Data 1 were far; therefore, Precision″ and MOTA were high, and IDSW was low. The distances between the fish in the group of Data 2 were close; therefore, Precision″ and MOTA were low, and IDSW was high. For both data sets, the best results were obtained when the fish-over-line rate threshold was 0% (no removal). However, increasing the fish-over-line rate threshold reduced point confusion with other individuals in the 3D position estimation and reduced IDSW. Table 6 shows the rate results of estimating the hidden fish's 3D position when occlusion occurs in one camera view (only those whose feature points could be estimated in the other two camera views). In Table

TABLE 6. Rate results of estimating the hidden fish's 3D position when occlusion occurs in one camera view (only those whose feature points could be estimated in the other two camera views)

Data	Threshold	Rate of hidden fish whose 3D positions could be estimated
2	0% (no removal)	1.000
	60%	–
	90%	–

TABLE 7. Rate results of the mirror image remaining after mirror-image elimination

Data	Threshold	Rate of mirror images remaining after 3D position estimation	Rate of mirror images remaining after mirror-image elimination
1	0% (no removal)	0.100	0.000
	60%	0.054	0.000
	90%	0.041	0.000
2	0% (no removal)	0.239	0.000
	60%	0.159	0.000
	90%	0.000	0.000

6, occlusion occurred only in Data 2, and the fish-over-line rate thresholds of 60% and 90% are left blank because hidden fish feature points were removed. The results in Table 6 show that even if fish occlusion occurred in one camera view, the 3D position of the hidden fish could be estimated if the feature points were estimated in the other two camera views. Table 7 shows the rate results of the mirror image remaining after mirror-image elimination in Data 1 and 2, respectively. The results in Table 7 show that some of the mirror images were eliminated after the 3D position estimation and that there were no mirror images remaining after the elimination of the mirror images.

5. **Conclusions.** Compared to the previously proposed two-camera method, the three-camera method of this study was able to estimate more 3D positions. Furthermore, even if fish occlusion occurred in one direction, 3D position estimation could be performed if feature points were estimated in the other two directions using the proposed method. Mirror images were eliminated, and the actual fish could be tracked, it can eliminate mirror images and track actual fish. This study solves the fish occlusion and mirror-image problems in a small tank and is expected to be applied to other studies dealing with animals in a small tank. For future work, the first is to improve the incorrectly assigned feature point removal method. This is because, although it can remove the incorrectly assigned feature points, it may also remove the correctly assigned feature points. The second is to improve the fish matching method. This was because, when the distance between the fish was close, the fish could not be narrowed down by matching and their 3D positions could not be estimated. The third is to improve the tracking method. This is because, although it uses the fish with the shortest Euclidean distance between the centroids of the feature points as tracking targets, if the fish cross each other, there is a possibility that the tracking targets may be switched.

REFERENCES

[1] T. Arimoto, *Why Fish Swim in Schools*, Taishukan Publishing, 2007.

[2] H. Yamaguchi, K. Tateno and K. Horio, 3D position estimation of multiple fish in a small tank using DeepLabCut and shortest distance calculation between two straight lines, *ICIC Express Letters*, vol.18, no.8, pp.835-842, 2024.
[3] M. G. B. Palconit, R. S. Concepcion II, J. D. Alejandrino, M. E. Pareja, V. J. D. Almero, A. A. Bandala, R. R. P. Vicerra, E. Sybingco, E. P. Dadios and R. N. G. Naguib, Three-dimensional stereo vision tracking of multiple free-swimming fish for low frame rate video, *Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII)*, vol.25, no.5, pp.639-646, 2021.
[4] R. Verschae, H. Kawashima and S. Nobuhara, A multi-camera system for underwater real-time 3D fish detection and tracking, *Proc. of the OCEANS Anchorage*, pp.1-5, 2017.
[5] R. Kawahara, S. Nobuhara and T. Matsuyama, A pixel-wise varifocal camera model for efficient forward projection and linear extrinsic calibration of underwater cameras with flat housings, *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp.819-824, 2013.
[6] O. Yamanaka and R. Takeuchi, UMATracker: An intuitive image-based tracking platform, *Journal of Experimental Biology (JEB)*, vol.221, no.16, 2018.
[7] J. M. Graving, D. Chae, H. Naik, L. Li, B. Koger, B. R. Costelloe and I. D. Couzin, DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning, *eLife*, vol.8, 2019.
[8] F. R. Ferrero, M. G. Bergomi, R. C. Hinz, F. J. H. Heras and G. G. de Polavieja, idtracker.ai: Tracking all individuals in large collectives of unmarked animals, *Nature Methods*, vol.16, pp.179-182, 2019.
[9] T. D. Pereira, N. Tabris, A. Matsliah, D. M. Turner, J. Li, S. Ravindranath, E. S. Papadoyannis, E. Normand, D. S. Deutsch, Z. Y. Wang, G. C. M. Smith, C. C. Mitelut, M. D. Castro, J. D'Uva, M. Kislin, D. H. Sanes, S. D. Kocher, S. S. H. Wang, A. L. Falkner, J. W. Shaevitz and M. Murthy, SLEAP: A deep learning system for multi-animal pose tracking, *Nature Methods*, vol.19, pp.486-495, 2022.
[10] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis and M. Bethge, DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning, *Nature Neuroscience*, vol.21, pp.1281-1289, 2018.
[11] T. Nath, A. Mathis, A. C. Chen, A. Patel, M. Bethge and M. W. Mathis, Using DeepLabCut for 3D markerless pose estimation across species and behaviors, *Nature Protocols*, vol.14, pp.2152-2176, 2019.
[12] J. Lauer, M. Zhou, S. Ye, W. Menegas, S. Schneider, T. Nath, M. M. Rahman, V. D. Santo, D. Soberanes, G. Feng, V. N. Murthy, G. Lauder, C. Dulac, M. W. Mathis and A. Mathis, Multi-animal pose estimation, identification and tracking with DeepLabCut, *Nature Methods*, vol.19, pp.496-504, 2022.
[13] P. Karashchuk, K. L. Rupp, E. S. Dickinson, E. Sanders, E. Azim, B. W. Brunton and J. C. Tuthill, Anipose: A toolkit for robust markerless 3D pose estimation, *Cell Reports*, vol.36, no.13, 2021.
[14] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler and B. Schiele, DeeperCut: A deeper, stronger, and faster multi-person pose estimation model, *European Conference on Computer Vision (ECCV)*, pp.34-50, 2016.
[15] K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, *IEEE/CVF Conferene on Computer Vision and Pattern Recognition (CVPR)*, pp.770-778, 2016.
[16] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li and L. F. Fei, ImageNet: A large-scale hierarchical image database, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.248-255, 2009.
[17] A. Mathis, T. Biasi, S. Schneider, M. Yüksekgönül, B. Rogers, M. Bethge and M. W. Mathis, Pretraining boosts out-of-domain robustness for pose estimation, *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp.1859-1868, 2021.
[18] M. Tan and Q. V. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, *The 36th International Conference on Machine Learning (PMLR)*, pp.6105-6114, 2019.
[19] Z. Zhang, A flexible new technique for camera calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol.22, no.11, pp.1330-1334, 2000.
[20] Open Computer Vision Library, *OpenCV*, https://opencv.org, Accessed on Oct. 23, 2024.
[21] OpenCV v2.1 Documentation, *Camera Calibration, and 3D Reconstruction*, http://opencv.jp/opencv-2.1_org/c/camera_calibration_and_3d_reconstruction.html, Accessed on Oct. 23, 2024.
[22] S. Ikehata, Y. Ushiku, Y. Utsumi, S. Ono, H. Kataoka, A. Kanezaki, Y. Kawanishi, M. Saito, K. Sakurada, K. Takahashi and Y. Matsui, *Computer Vision –Expanding Elemental Technologies and Applications–*, Kyoritsu Publishing, 2018.
[23] OpenCV 2 Programming Book Production Team, *OpenCV2 Programming Book*, Mynavi, 2011.

[24] Digital Image Processing Editorial Board, *Digital Image Processing*, 2nd Edition, CG-ARTS, 2020.
[25] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man, and Cybernetics (SMC)*, vol.9, no.1, pp.62-66, 1979.
[26] N. Sannomiya, Behavior model and simulation for fish school, *Fisheries Engineering*, vol.30, no.1, pp.41-47, 1993.
[27] Imgaug 0.4.0 Documentation, *Imgaug*, https://imgaug.readthedocs.io/en/latest/, Accessed on Jan. 28, 2025.

Author Biography

Hiroki Yamaguchi received his B.E. degree from Nishinippon Institute of Technology, Japan in 2020 and his M.E. degree from Kyushu Institute of Technology, Japan in 2022. He is currently a Ph.D. student at Kyushu Institute of Technology, Japan. His research interests include behavior analysis.

Katsumi Tateno received his B.E., M.E., and Ph.D. degrees from the Kyushu Institute of Technology, Japan, in 1994, 1996, and 1999, respectively. He was a postdoctoral fellow at McGill University, Montreal, Canada, from 1999 to 2002. He was an Assistant Professor at the Kyushu Institute of Technology, Japan from 2002 to 2005. He is currently a Professor at the Kyushu Institute of Technology, Japan. His research interests include neuronal dynamics in hippocampal neural networks.

Keiichi Horio received his B.E., M.E., and Ph.D. degrees from the Kyushu Institute of Technology, Japan, in 1996, 1998, and 2001, respectively. He is currently a Professor at the Kyushu Institute of Technology, Japan. His research interests include behavior analysis and intelligent information processing.