Sensors and Materials, Vol. 32, No. 10 (2020) 3491–3506
MYU Tokyo

S & M 2353

Lead-lag Swimming Control of Robot Fish Using Color Detection Algorithm

Amarnathvarma Angani, Seung Hoon Kang, Teressa Talluri, Tae Uk Kang, Myeong Jin Seo, and Kyoo Jae Shin*

Department of ICT Creative Design, Busan University of Foreign Studies,
65, Geumsaem-ro 485 Beon-gil, Geumjeong-gu, Busan 46234, South Korea

(Received April 17, 2020; accepted August 14, 2020)

Keywords: robot fish, lead-lag swimming control, image processing, object detection algorithm

This paper is about lead-lag robot fish swimming control by image processing. In robot fish aquariums, fish can generally move randomly in any path. This irregular movement may cause collisions between the fish, damaging them. Hence, an effective swimming control method is necessary. We thus proposed a lead-lag swimming control system for robot fish. Here, we simply study the detection of moving objects in an aquarium because we need to find the positions of moving robot fish. The locations of the robot fish are recognized using an image processing technique employing image and position sensors. This method is used to obtain the velocity for each pixel in an image and assumes a constant velocity in each video frame to obtain the positions of robot fish by comparing consecutive video frames. By using the position data, we compute the distance between robot fish and determine which robot fish is the lead fish and which fish is the lagging fish. The lead fish then waits for the lagging fish to catch up. The results of this proposed system are satisfactory in preventing collisions between robot fish. This system is exhibited in Busan Science Museum in South Korea.

1. Introduction

Nowadays, robotic devices are used to study animal behavior. Many researchers have attempted to reproduce the movements of animals such as cockroaches, chicks, and honeybees.[1] Recently developed bio-inspired robot fish are imitating various aspects of nature. Often, engineers get ideas from biological systems, inspiring them to make new biomimetic robots. Several robot fish have been developed by researchers such as those from MIT, who extended their research on implementing robot fish with various algorithms.[2,3] Extensive research has been carried out on fish behavior underwater. In earlier studies on robot fish, the aspect ratios of robot zebrafish movement were traced out.[4] Hirata *et al.*[5] developed robot fish with various abilities that were inspired by boats and submarines. Researchers from the University of Essex attempted to develop autonomous robot fish that swam like real fish.[6] Worm *et al.*[7] found that a mobile robot that emitted weak electrical signals can be used to attract electric

*Corresponding author: e-mail: kyoojae@bufs.ac.kr
https://doi.org/10.18494/SAM.2020.2926

ISSN 0914-4935

fish. Shin and coworkers[8,9] developed a method of controlling robot fish movement with hand movements. Several methods have been adopted to reduce the motion of real fish by robot fish. For instance, fish locomotion has been modeled with the jump persistent turning walker model inspired by sudden changes in zebrafish locomotion in the form of large deviations.[10] Zienkiewic *et al.*[11] applied a stochastic model to zebrafish in a specific environment. Although robotic devices that can interact with real fish have been developed, there have been no studies on how to control the paths of two or three robot fish moving along the same path but in opposite directions. Because fish can move in any path in water, if there is no path detection or path control, collisions may occur, resulting in a time-consuming process required to reset the path of the fish and the rapid discharge of the battery of the robot fish.

To avoid this problem, here, we propose a new method of lead-lag swimming control for robot fish with position-detecting sensors and algorithms. The robot fish used in this study were modeled on the *domi* fish, which lives in the sea offshore of South Korea. The main aim of this study is to develop a method of controlling the motion of all the robot fish in an aquarium with the lead-lag swimming control mechanism to ensure the collision-free and uniform movement of two or three robot fish following the same path.

Sensors are important components in tracking the movements of robot fish and guiding them along the correct path without collisions. We detected and tracked the positions of robot fish using different algorithms such as boundary, optical flow, and color segment algorithms. We analyzed the performance of the three algorithms and adopted the most suitable algorithm for this system. Finally, we tracked the motion of robot fish using various tracking algorithms such as stop-zone tracking control, which is based on manual position commands to stop the robot fish, and color-mark tracking control, which is based on color marks placed on the sides of the aquarium, where the robot fish stop after reaching them.

Section 2 of this paper describes the methodology of lead-lag swimming control and robot fish construction, Sect. 3 describes the algorithms that were applied through the OpenCV computer library to perform tasks. Sections 4 and 5 describe the components of our experimental setup, such as the hardware and software, and the results, respectively. Finally, we give conclusions in Sect. 6. The performance of lead-lag swimming control was excellent, and the robot fish employing the control method were exhibited in the 2012 International YEOSU EXPO.

2. Methodology of Lead-lag Swimming Control

The working principle of the proposed method of lead-lag swimming control is that all fish in an aquarium move uniformly with uniform velocity in a group without collisions. The lead fish should wait for the lag fish to catch up, after which the two fish can swim without collisions. To accomplish this, various sensors are used to detect color and position, and radio frequency (RF) ID sensors are used to track the robot fish. This lead-lag mechanism is implemented using a control algorithm. This algorithm mainly consists of four blocks, namely, a video acquisition block, a color detection block, a robot fish motion block, and a serial port block, as shown in Fig. 1. In the video acquisition block, a video is acquired from a camera,

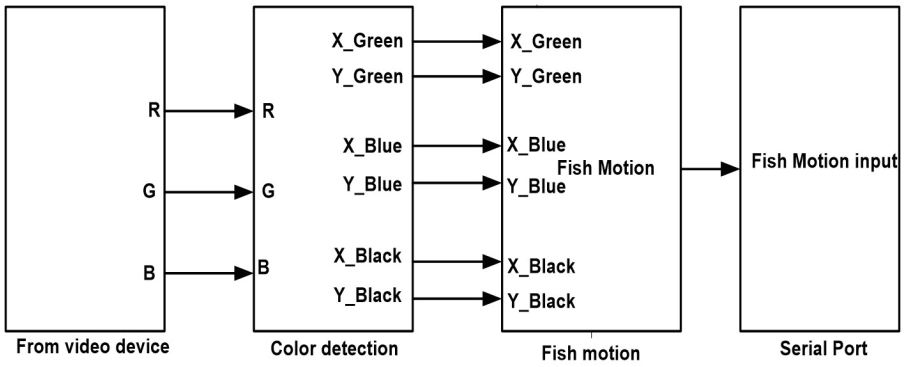


Fig. 1. Layout for lead-lag control of robot fish.

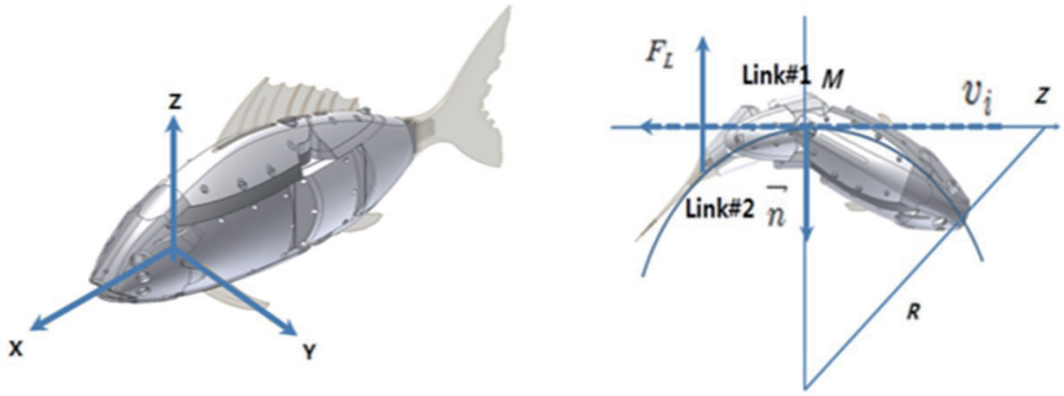
and objects are detected using a color segment algorithm. Through this algorithm, we applied different vivid colors, such as green, blue, and black, to different robot fish to detect them. This detection generates coordinates, making it easy to evaluate the positions of the robot fish. By using these position coordinates (x, y), we can compute the distance between the robot fish, allowing the lead and lagging robot fish to be identified. Then, if the distance between the two fish is more than 150 mm, the lead fish will wait for the lagging fish to catch up. This involves sending control commands to the fish, such as forward, left, right, up, and down, via an RF modem through a serial port. If the distance is less than 150 mm, all robot fish swim in the autonomous mode. The robot fish were developed as biomimetic fish for use in an aquarium.

The robot fish consist of a head, a first-stage body, a second-stage body, and a tail, which are connected through joints. The body of the robot fish is designed through the analysis of biological fish swimming to maximize its momentum.

The control system consists of three position-sensitive detector (PSD) sensors, two DC motors, a weight moving unit, and an RF modem. The sensors are interactive components used to detect the position of the robot fish and send signals to the proportional-integral-derivative (PID) controller to control its motion. The instantaneous dynamic force acting on the robot fish is determined from the thrust (in the forward direction or x-axis), water resistance (in the reverse direction or y-axis), and gravity (in the vertical direction or z-axis) acting on the robot fish as shown in Fig. 2(a). When the robot fish is swimming, the force acting on its body is the gravitational force, which acts opposite to the direction of fluid propulsion of the fish, lifting its body as shown in Fig. 2(b).

Motor speed is an important parameter for robot fish movement. Hence, the speed of the motor should be controlled for smooth and safe swimming. The motor speed depends on the torque value T_L. To decrease motor speed, torque should be reduced. The performance of controlling the speed of the motor is improved by using a proportional feedback controller. The controller is composed of a sensor to sense the speed and an amplifier with gain K (proportional control). The speed at the motor shaft is sensed by a potentiometer with gain K_t and a PID controller.[12,13]

The Simulink model used to control the links in the body of the robot fish is shown in Fig. 3. The link angle output (LAO) is expressed as



(a) (b)

Fig. 2. (Color online) (a) Axes of the robot fish and (b) lift and swimming rotation in an aquarium.

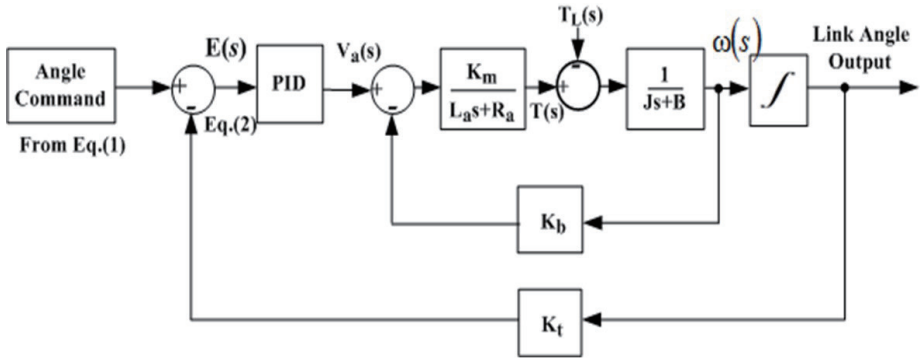


Fig. 3. Link angle control of the robot fish.

$$LAO = \frac{s^2\ K_m K_d + s\ K_i K_p + K_i K_m}{s^3\ R_a\ J + s^2(K_m\ K_b + R_a B + K_m\ K_t\ K_d) + s(K_m\ K_t\ K_p) + K_m\ K_t\ K_i}, \quad (1)$$

where K_m is the motor gain, K_d is the differential gain, K_i is the integral gain, K_p is the proportional gain, J is the moment of inertia of the motor, K_b is the gain of the back emf, K_t is the angle signal gain, R_a is the armature resistance, B is the viscous friction coefficient of the motor shaft, and L_a is the armature inductance.

3. Lead-lag Swimming Control Algorithms

The proposed concept of lead-lag swimming control can be implemented by applying four algorithms: a boundary detection algorithm to detect edges, an optical flow algorithm to estimate the speed and direction of the robot fish, a color segment algorithm to recognize color, a lead-lag robot fish position detection algorithm, and a lead-lag tracking control algorithm. These algorithms are explained in detail in the following sections.

3.1 Boundary detection algorithm

To find the position of the robot fish, its boundary must first be detected. This involves four stages: filtering, enhancement, detection, and localization. To detect the edges of the robot fish, we used the Roberts, Prewitt, and Sobel operators as edge detectors. The process for edge detection is shown in Fig. 4.

In these algorithms, image processing is used to detect the boundary. This algorithm is developed using boundary detectors. The Sobel operator is very similar to the Prewitt operator. It is also used for boundary detection. Similarly to the Prewitt operator, the Sobel operator is also used to detect two types of edges in an image: those in the vertical direction and those in the horizontal direction.

(1) Roberts operator: The Roberts operator computes a simple approximation of the magnitude of the gradient of the pixels of an image.

(2) Prewitt operator: The Prewitt operator introduces 3 × 3 gradient operators by using a pixel numbering convention. The Prewitt operator square root edge gradient is shown in Eq. (2), where $k = 1$. In this process, the row and column gradients are normalized to provide unit-gain positive and negative weighted averages for separate edge positions.

$$G(j,k) = \left[[G_R(j,k)]^2 + [G_C(j,k)]^2 \right]^{1/2} \tag{2}$$

Here, $G(j, k)$ is the spatial gradient amplitude and $G_R(j, k)$ is the row gradient.

$$\begin{aligned} G_R(j,k) &= \frac{1}{K+2}\left[(A_2 + KA_3 + A_4) - (A_0 + kA_7 + A_6)\right], \\ G_C(j,k) &= \frac{1}{K+2}\left[(A_0 + KA_1 + A_2) - (A_6 + kA_5 + A_4)\right]. \end{aligned} \tag{3}$$

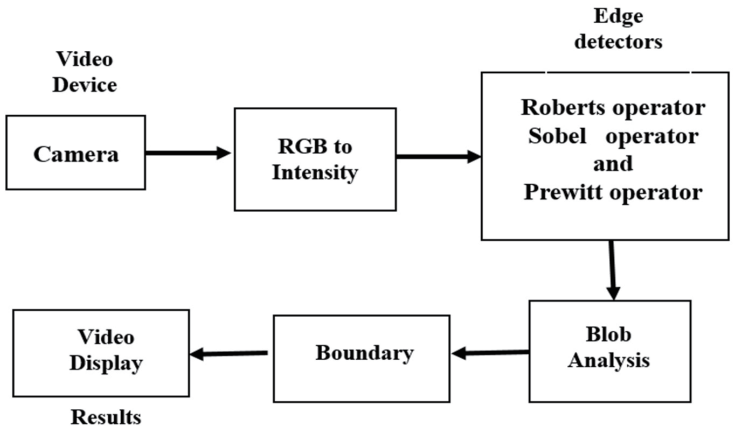


Fig. 4. Process of edge detection using Roberts, Prewitt, and Sobel operators.

(3) Sobel operator: The Sobel operator edge detector differs from the Prewitt operator edge detector in that the values of the north, south, east, and west pixels are doubled when k is 2. This operator can be expressed using Eq. (3), where k is a constant and A_0, A_1, A_2, A_3, A_4, A_5, and A_6 are the neighborhood pixels.

3.2 Optical flow algorithm

The optical flow algorithm is used to analyze the motion of the moving objects and estimate the direction and speed of an object from one image to another or from one video frame to another as shown in Fig. 5.

The optical flow algorithm is based on the frames of a moving object. It compares two frames and estimates the motion between them to find the position of an object because every object has some pixels that do not change when the image is captured by a camera, allowing the position of the robot fish to be detected. For instance, in a real-time video, consider two frames, frames $n - 1$ and n, and their times t and $t + 1$ s, respectively. When we compare frames $n - 1$ and n, the position of the robot fish will be changed, and the position of the object is detected with the optical flow algorithm. The standard equation of an optical flow used to detect the motion between frames in a video and obtain the motion vectors on the object is given by

$$f_x u + f_y v + f_t = 0\,, \tag{4}$$

where $f_x = \frac{\partial f}{\partial x}, f_{y,} = \frac{\partial f}{dy}, f_t = \frac{\partial f}{\partial t}, \frac{dx}{dt} = u, \frac{dy}{dt} = v.$

From Eq. (4), we can find the image gradients f_x and f_y and the gradient with respect to time, f_t. In Eq. (4), u and v are respectively the horizontal and vertical directions of the optical flow, which are also known as motion vectors.[14–17] To compute the motion vectors, the optical flow

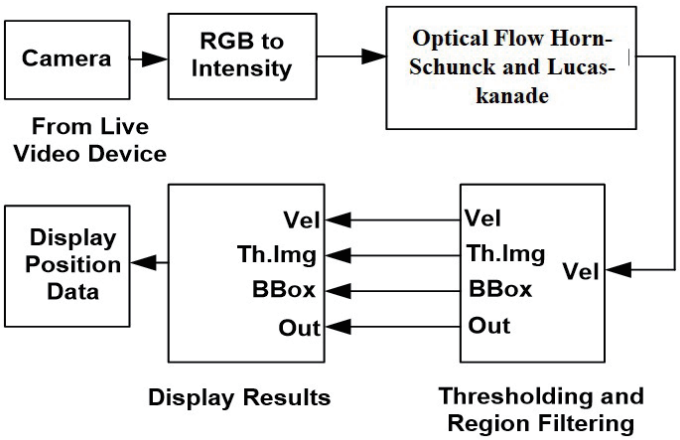


Fig. 5. (a) Frames of robot fish and (b) layout of optical flow process.

limitation equation [Eq. (5)] is used with the Lucas–Kanade method. From this equation, we assume that neighboring pixels will have similar motions. This method takes a 3 × 3 patch around each point. For all the points to have the same motion, we find (f_x, f_y, f_t) for these nine points. From these points, we can find the motion vectors (u v).

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i f_{x_i}{}^2 & \sum_i f_{x_i} f_{y_i} \\ \sum_i f_{x_i} f_{y_i} & \sum_i f_{y_i}{}^2 \end{bmatrix} \begin{bmatrix} -\sum_i f_{x_i} f_{t_i} \\ -\sum_i f_{y_i} f_{t_i} \end{bmatrix} \tag{5}$$

The optical flow process shown in Fig. 5(b) is explained as follows:

(1) Live video device: To set the camera, double click on this block → select the camera → select the Apply button → select OK. The output of this block is the image input to the second block, that is, RGB to intensity.
(2) RGB to intensity: This block takes the color data and converts it into the gray color by using color codes from 000 to 255.
(3) Optical flow: This optical flow is the pattern of the apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and the scene. In this block, the velocity of each frame is estimated by using an optical flow method. One of the two methods can be selected in this block: the Horn–Schunck or Lucas-Kanade optical flow method.
(4) Thresholding and filtering: This block thresholds and filters the image from the original image. This block locates the fish in each binary image using the blob analysis block. Then, it uses the draw shapes block to draw a green rectangle around the robot fish.
(5) Display results: The results window shows the position of the fish in the green rectangle in the region of the aquarium.
(6) Display position data: This window shows the position data of the detected robot fish in the aquarium.

3.3 Color segment algorithm

The color segment algorithm is shown in Fig. 6. It depicts the object detected, fish position, and display from the USB camera. This algorithm starts from the USB camera, which uses the HSV to BGR color space to detect vivid color objects. The next step is to declare the arguments in the algorithm of OpenCV. There are three arguments: the first one is the source image, the second one is the contour retrieval mode, and the third one is the contour approximation method. This algorithm outputs the contours and hierarchy, where the former is a Python list of all the contours in the image.

Each contour is a NumPy array of (x, y) coordinates of boundary points of the object. Contours can be explained simply as a curve joining all the continuous points (along the boundary) having the same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition. Ultimately, the position of the robot fish will be displayed on a display block.

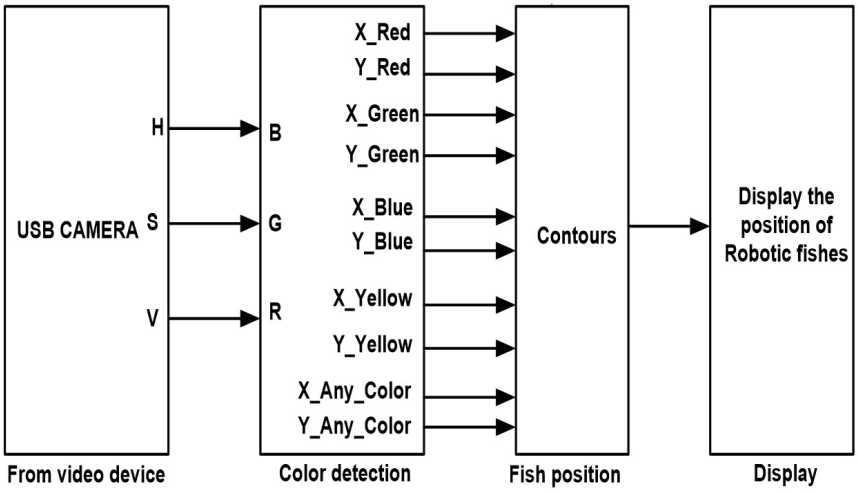


Fig. 6. Color segment algorithm using OpenCV.

3.4 Lead-lag tracking control algorithm

Lead-lag control using the color segment algorithm is applied to the robot fish to detect the position using its color. Using this position data, we compute the distance between robot fish using Eq. (6) and determine which robot fish is leading and which robot fish is lagging. Then, the lead fish waits for the lagging fish to catch up. If the distance becomes 150 mm, then the robot fish move parallel to each other. This process is repeated.

$$d_1 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{6}$$

Here, d_1 is the distance between robot fish 1 and robot fish 2, and x_1 and x_2 are the x-coordinates and y_1 and y_2 are the y-coordinates of robot fish 1 and robot fish 2, respectively. After evaluating the distance, the algorithm for lead-lag tracking control shown in Fig. 7 is applied to the lagging fish.

A comparison of the parameters of the three algorithms developed in this study is shown in Table 1. We concluded that the color segment algorithm is most suitable for this system to track the robot fish in the aquarium.

4. Experimental setup of lead-lag swimming control method

The experimental setup for controlling robot fish in an aquarium consists of a personal computer (PC), a camera, robot fish, and a RF modem as shown in Fig. 8(a). The proposed robot fish tracking position control system consists of control of the link in the body of the robot fish and the optical flow detection algorithm as shown in Fig. 8(b). After detecting the moving object, we obtain its position data. The RF modem communicates the robot fish position data between the PC and the robot fish. By using angle commands, the robot fish can turn left, right, forward, up, and down.

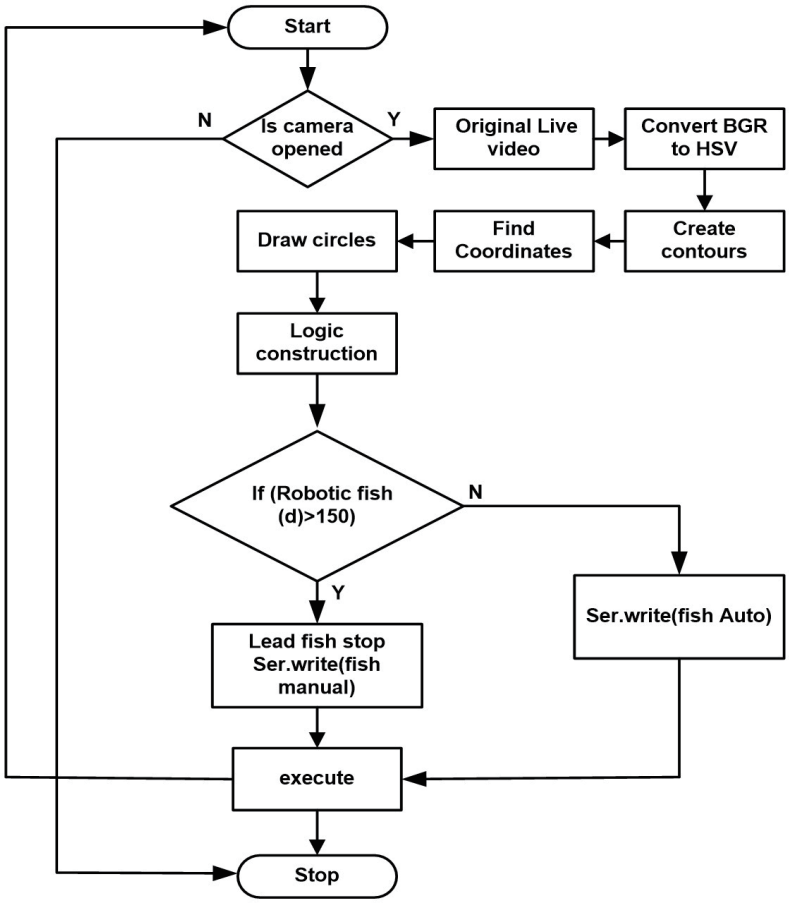


Fig. 7. Flow chart of the lead-lag tracking control.

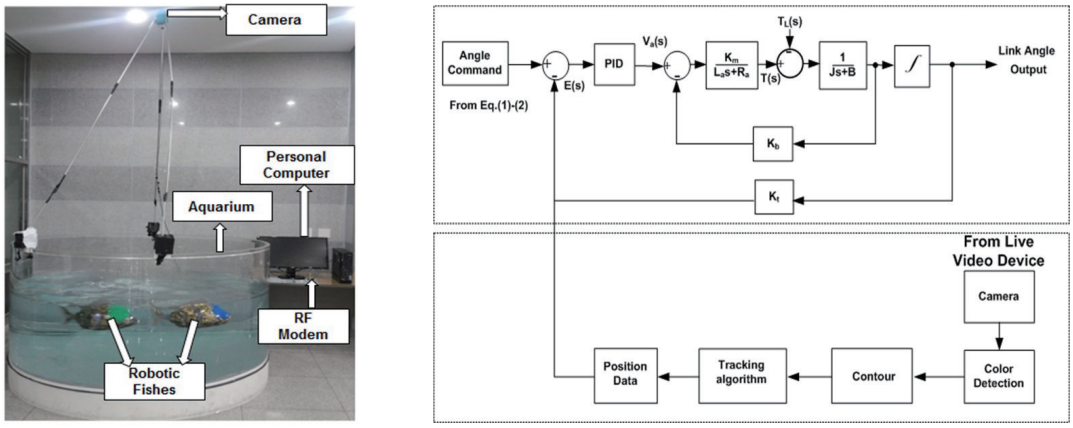
Table 1
Comparison among the object detection algorithms for robot fish.

Parameter	Boundary algorithm	Color segment algorithm	Optical flow algorithm
Object detection	Edge detectors (Sobel, Roberts, Prewitt)	Color threshold depends on color	Lucas–Kanade optical flow and dense optical flow
Threshold	Edges	Colors	Motion vectors
Variability	Detects all objects	Detects each object	Detects all objects
Position data	Not accurate	Good	Moderate
Identification of object	Complex	Simple	Complex
Noise	Yes	No	Yes
Accuracy	Not stable	Stable	Not stable
Tracking	Not possible	Possible	Possible with noise

5. Experimental Results of Algorithms

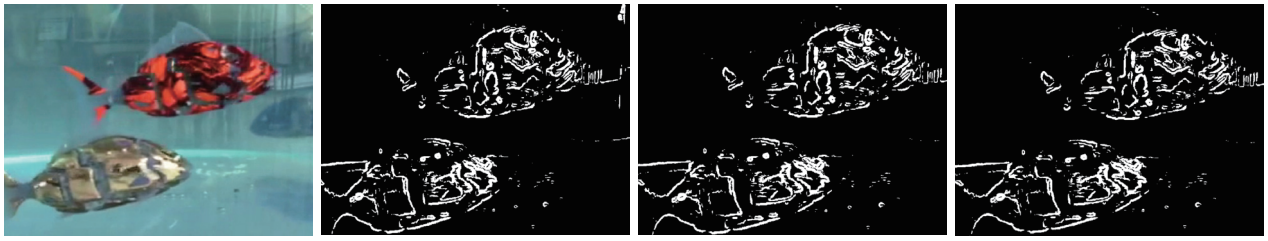
5.1 Result of boundary detection algorithm

Figure 9(a) shows the original object source directly captured by the side-view camera and Fig. 9(b) shows the edges of the robot fish detected using the Roberts edge detector. This edge operator uses a large mask to reduce the noise from the image.



(a) (b)

Fig. 8. (Color online) (a) Experimental setup of lead-lag swimming control and (b) proposed robot fish position control system.



(a) (b) (c) (d)

Fig. 9. (Color online) Detection of robot fish. (a) Original video, (b) edges detected using Roberts edge detector, (c) edges detected using Sobel edge detector, and (d) edges detected using Prewitt edge detector.

Figure 9(c) shows the edges of the robot fish detected using the Sobel edge detector. Figure 9(d) shows the edges of the same robot fish detected using the Prewitt edge detector. Similar results were obtained for both detectors; the main difference between the detectors is the weight of the mask or kernel. In fact, the Roberts, Sobel, and Prewitt edge detectors all gave similar results.

The HSV color space belongs to the group of hue-oriented color-coordinate systems. This type of color model closely emulates models of human color perception. While in other color models, such as RGB, an image is treated as an additive result of three base colors, the three channels of HSV represent hue (*H* gives a measure of the spectral composition of a color), saturation (*S* gives the proportion of pure light of the dominant wavelength, which indicates how far a color is from a gray of equal brightness), and value (*V* gives the brightness relative to that of a similarly illuminated white color), corresponding to the intuitive features of tint, shade, and tone. The color space conversion codes to convert between RGB and HSV in OpenCV using cvtColor are COLOR_BGR2HSV, COLOR_RGB2HSV, COLOR_HSV2BGR, and COLOR_HSV2RGB. In this case, note that if the source image format is 8-bit or 16-bit, cvtColor first converts it to a floating-point format with the values scaled between 0 and 1.

After that, the transformations are computed from Eqs. (7) and (8). If $H < 0$, then $H = H + 360$. Finally, the values are reconverted at the destination.

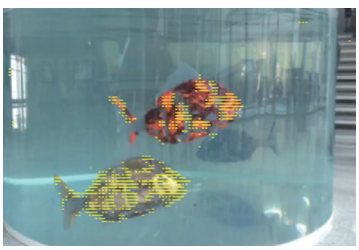
$$V = \max(R, G, B)$$
$$S = \begin{cases} \dfrac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$H = \begin{cases} \dfrac{60(G - B)}{V - \min(R, G, B)} & \text{if } V = R \\ 120 + \dfrac{60(B - R)}{V - \min(R, G, B)} & \text{if } V = G \\ 240 + \dfrac{60(R - G)}{V - \min(R, G, B)} & \text{if } V = B \end{cases} \tag{8}$$

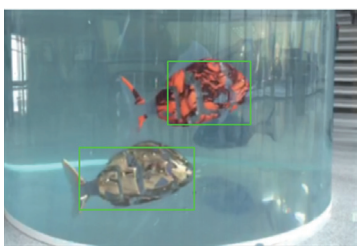
5.2 Result of optical flow algorithm

The experiment of this algorithm was held in two camera views using front view and top view cameras. The experimental results of the optical flow algorithm using the front view camera are shown in Fig. 10. Let f4 be the red robot fish and f5 be the yellow robot fish, whose coordinates are (x_1, y_1) and (x_2, y_2) respectively, as shown in the results. Figure 10(a) shows the identification for the position of robot fish with motion vectors. Motion vectors around the robot fish will give the coordinates to identify. Figure 10(b) shows that the thick green color rectangle is surrounded by the robot fish that indicates the detection of the robot fish in the aquarium and Fig. 10(c) shows that the robot fish coordinates after detecting the moving object are (x_1, y_1: 719, 528), (x_2, y_2: 689, 564).

By using this position data, we send the control data to control and track the robot fish using the RF modem. The RF modem can send and receive the data because it has both transmitter (TX)



(a)



(b)

No.	*X*	*Y*
f4	719	528
f5	689	564

(c)

Fig. 10. (Color online) Results of optical flow algorithm: (a) motion vector of robot fish and (b) fish detected from the optical flow, and (c) coordinates of robot fish.

and receiver (RX) modules included in the single unit with low carrier frequency, from or to the PC. When we compare the performance characteristics of the two views of the camera, the top view camera is found to detect the waves of the water and make some noise while detecting the object.

5.3 Results of color detection algorithm

Figure 11(a) shows the robot fish covered by the camera and Figs. 11(b)–11(e) show the results of the color segment algorithm, obtained from the various colors of robot fish. Here, we threshold an object to filter in the water to detect each object and obtain the position data that we will apply to the tracking control algorithms to track each robot fish at a particular position in the aquarium.

5.4 Results of lead-lag swimming control algorithm

Figure 12(a) shows an image of two robot fish taken from the live video before lead-lag control. The lead fish is green and the lagging fish is blue. These two robot fish are swimming counterclockwise in the aquarium. Figure 12(b) shows the motion of the robot fish after lead-lag control. When the distance between two robot fish is more than 150 mm, the lead robot fish waits for the lagging robot fish until it reaches the lead robot fish. The results are highly satisfactory.

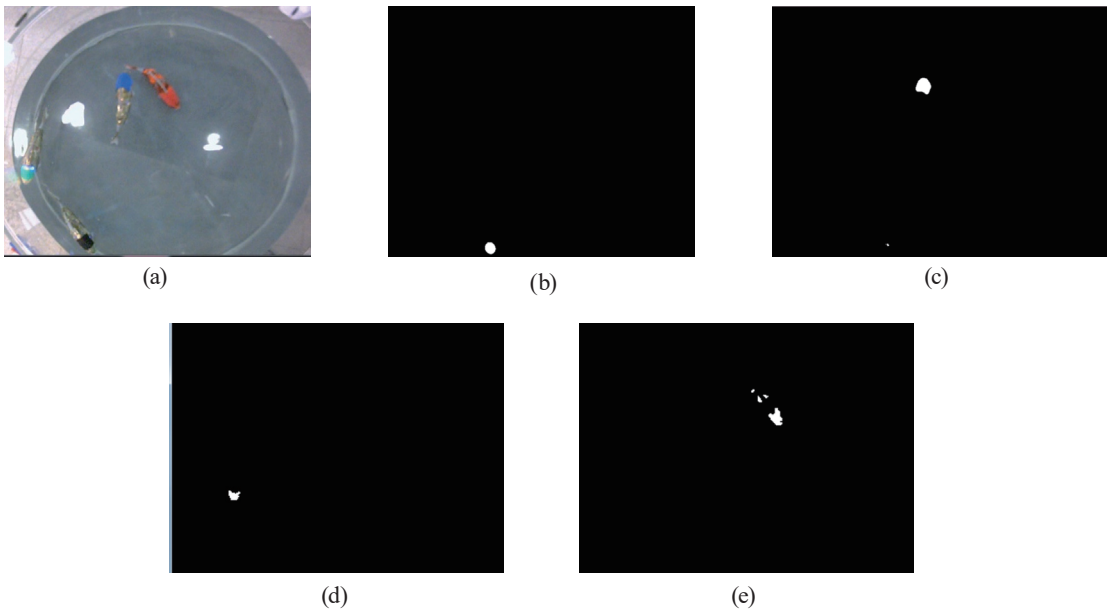


Fig. 11. (Color online) Results of color segment algorithm. (a) Robot fish in aquarium, (b) detecting black color robot fish, (c) detecting blue color robot fish, (d) detecting green color robot fish, and (e) detecting red color robot fish.

(a) (b)

Fig. 12. (Color online) Lead-lag swimming control. (a) Robot fish before lead-lag motion and (b) robot fish after lead-lag motion.

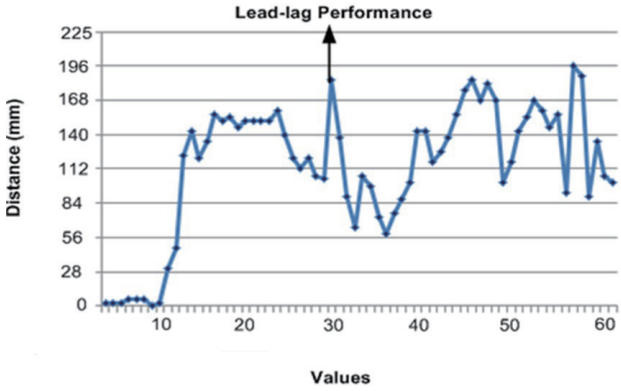


Fig. 13. (Color online) Lead-lag distance tracking between two robot fish.

5.5 Lead-lag tracking analysis of robot fish

The algorithm showed satisfactory performance in the test. To acquire data, a live video device was used. We then analyzed the video data by using the proposed methods. The distance between the two robot fish under lead-lag control is shown in Fig. 13. When the distance is less than 150 mm, the robot fish will continue to swim.

6. Conclusion

In this paper, we explained lead-lag swimming control by applying algorithms. To find the position of moving robot fish in an aquarium, we first have to detect them, for which we used an optical flow algorithm. We used a Python model to display the coordinates of the robot fish to analyze their position. Boundary detection and optical flow algorithms were applied to detect the boundaries and motion of the robot fish. The outcome of these algorithms is identifying

Fig. 14. (Color online) Robot fish tracking control system in Busan Science Museum.

the position of each robot fish. By using the position data, we computed the distance between the robot fish using position-detecting sensors and determined which robot fish is leading and which robot fish is lagging. The lead robot fish waits for the lagging fish until it reaches the lead robot fish. If the distance between them becomes 150 mm, then the robot fish move parallel to each other. The lead-lag swimming control method is used in a Python image processing model for tracking and detecting the robot fish. Excellent performance was observed in a field test conducted for the filtering and enhancement of an image, object detection, motion vectors, and the localization of the boundary to finally achieve the lead-lag control of the robot fish. Our lead-lag swimming control system has very good performance and is displayed in Busan Science Museum as shown in Fig. 14.

Acknowledgments

This work was supported by the Korea Ministry of Trade, Industry and Energy under the grant of the "Design Expert Training for Factory Automatic of the Based ICT Energy" project.

References

1 P. Gaikwad and P. Bansod: Int. J. Sci. Tech. Eng. **2** (2019) 1. https://pdfs.semanticscholar.org/d992/1d50c9cdfe763034abe69fd8ca84863ef0b9.pdf
2 Y. L. Maho, J. D. Whittington, N. Hanuise, L. Pereira, M. Boureau, M. Brucker, N. Chatelain, J. Courtecuisse, F. Crenner, B. Friess, and E. Grosbellet: Nat. Methods **11** (2014) 1242. https://www.nature.com/articles/nmeth.3173
3 J. M. Kumph: Master's Thesis, MIT (2000) pp. 1–76. https://dspace.mit.edu/bitstream/handle/1721.1/8968/47045662-MIT.pdf?sequence=2&isAllowed=y
4 N. Abaid, T. Bartolini, S. Macri, and M. Porfiri: Behav. Brain Res. **233** (2012) 545. https://doi.org/10.1016/j.bbr.2012.05.047
5 K. Hirata, T. Takimoto, and K. Tamura: Proc. 1st Int. Symp. Aqua Bio-Mechanics (2000) 287–289. https://www.nmri.go.jp/oldpages/eng/khirata/list/fish/isamec2000.pdf
6 H. Hu: Proc. IEEE SMC UK-RI Chapter Conf. Advances in Cybernetic Systems, Sheffield (2006) 1–8. https://cswww.essex.ac.uk/staff/hhu/Papers/AICS2006-InvitedTalk-Hu.pdf
7 M. Worm, T. Landgraf, H. Nguyen, and G. E. Emde: Proc. Conf. Biomimetic and Biohybrid Systems (Springer, Switzerland, 2014) 446–448. https://link.springer.com/chapter/10.1007/978-3-319-09435-9_57.
8 H. Kim, E. S. Kim, C. B., T. Teressa, A. Angani, and K. J. Shin: Proc. Institute of Electronics and Information Engineers (IEIE, 2019) 817–821.

9 A. V. Angani, J. H. Lee, and K. J. Shin: Proc. Korea Information Processing Society Conf. (2016) 1354–1357.
10 V. Mwaffo, R. P. Anderson, S. Butail, and M. Porfiri: J. R. Soc. Interface **12** (2015) 20140884. https://doi.org/10.1098/rsif.2014.0884
11 A. Zienkiewicz, D. A. Barton, M. Porfiri, and M. D. Bernardon: J. Math. Biol. **71** (2015) 1081. https://link.springer.com/article/10.1007/s00285-014-0843-2
12 K. J. Shin: KIPS Trans. Software Data Eng. **4** (2015) 219. https://doi.org/10.3745/KTSDE.2015.4.5.219
13 K. J. Shin: IEIE Trans. Smart Process. Comput. **5** (2016) 375. https://doi.org/10.5573/IEIESPC.2016.5.6.375
14 K. J. Shin and Y. M. Rao: ICEIC Danang, Vietnam **2** (2016) 81. https://doi.org/10.1109/ELINFOCOM.2016.7562987
15 S. H. Kang, T. U. Kang, A. Angani, and J. B. Lee: Proc. IEEE Int. Conf. Architecture, Construction, Environment and Hydraulics (IEEE, 2019) 21–24.
16 A. V. Angani, J. W. Lee, T. Teressa, J. Y. Lee, and K. J. Shin: Sens. Mater. **32** (2020) 3479. https://doi.org/10.18494/SAM.2020.2925
17 A. V. Angani, T. Teressa, J. W. Lee, E. S. Kim, and J. B. Lee: Sens. Mater. **32** (2020) 3507. https://doi.org/10.18494/SAM.2020.2927

About the Authors

Amarnathvarma Angani is pursuing a Ph.D. degree under the guidance of Professor Kyoo Jae Shin in the Department of ICT Creative Design at Busan University of Foreign Studies (BUFS), Busan, South Korea. He received his master's degree in robotics from BUFS. He obtained his B.Tech degree in mechanical engineering in 2014 from Jawaharlal Nehru Technological University, Kakinada, India, and his diploma in automobile engineering in 2007 from Andhra Polytechnic, Kakinada, India. He worked as a design engineer at InfoTech Enterprises from 2007 to 2011. His research deals with smart fish farms. He is also interested in control systems, fluid dynamics and thermodynamics, and vehicle design. (amarnath.angani@gmail.com)

Seung Hoon Kang is an elementary school teacher at Bumin Public Elementary School. He is pursuing a Ph.D. degree under the guidance of Professor Kyoo Jae Shin in the Department of ICT Creative Design at BUFS, Busan, South Korea. He received his master's degree in robotics engineering from BUFS in 2019 and his bachelor's degree from the Department of Early Childhood Education at Busan National University of Education, Busan, South Korea in 2011. His research deals with smart fish farms and reinforcement learning. He is also interested in software education and computational thinking. He worked as an SW teacher at Busan SW Education Support Center from 2018 to 2019. (chomin308@gmail.com)

Teressa Talluri is pursuing a Ph.D. degree in the Department of ICT Creative Design at BUFS, Busan, South Korea. She received her master's degree in thermal engineering from Jawaharlal Nehru Technological University, Kakinada, India, in 2014.

Tae Uk Kang is pursuing a master's degree in the Department of ICT Creative Design at BUFS, Busan, South Korea. (jimpro@naver.com)

Myeong Jin Seo is pursuing a bachelor's degree in the Department of ICT Creative Design at BUFS, Busan, South Korea. (tjaudwlswkd@naver.com)

Kyoo Jae Shin is a professor of intelligent robot science at BUFS, Busan, South Korea. He is the director of the Future Creative Science Research Institute at BUFS. He received his B.S. degree in electronics engineering in 1985, his M.S. degree in electrical engineering from Cheonbuk National University (CNU) in 1988, and his Ph.D. degree in electrical science from the Pusan National University in 2009. Dr. Shin was a professor at a navy technical education school and a senior director of a research association of dynamic stabilization systems in a defense weapon research institute in Dusan. He has also researched and developed robot fish, submarine robots, automatic milking robot using a manipulator, a personal electrical vehicle, a smart accumulated aquarium using a heat pump, a solar tracking system, a 3D hologram system, and a gun/turret stabilization system. He is interested in intelligent robots, image signal processing application systems, and smart farms and aquariums using new energy and Internet of Things (IoT) technology.