A Study of Detecting The Fish Robot Position Using The Object Boundary Algorithm

Amarnath Varma Angani*, Kang Min Jeong**, Shin Kyoo Jae*
*Depart of ICT Creative Design, Graduate School, M.S.Course
**Depart of Digital Media Engineering, B.S.Course
Busan University of Foreign Studies
Busan, Republic of Korea
e-mail : amarnath.angani@gmail.com, lucky6990@nate.com, kyoojae@bufs.ac.kr

물체 형상인식 알고리즘을 이용한 물고기 로봇 위치 검출에 관한 연구

아마르나 바르마 앙가니*, 강민정**, 신규재*
*부산외국어대학교 일반대학원, ICT 창의융합학과
**부산외국어대학교 디지털미디어공학부
e-mail : amarnath.angani@gmail.com, lucky6990@nate.com, kyoojae@bufs.ac.kr

ABSTRACT

In this paper, we have researched about how to detect the fish robot objects in aquarium. We had used designed fish robots DOMI ver1.0, which had researched and developed for aquarium underwater robot. The model of the robot fish is analysis to maximize the momentum of the robot fish and the body of the robot is designed through the analysis of the biological fish swimming. We are planned to non-external equipment to find the position and manipulated the position using creating boundary to fish robot to detect the fish robot objects. Also, we focused the detecting fish robot in aquarium by using boundary algorithm. In order to the find the object boundary, it is filtering the video frame to picture frames and changing the RGB to gray. Then, applied the boundary algorithm stand of equations which operates the boundary for objects. We called these procedures is kind of image processing that can distinguish the objects and background in the captured video frames. It was confirmed that excellent performance in the field test such as filtering image, object detecting and boundary algorithm.

1. Introduction

The fish robot had studied various areas for Hardware (H/W) and Software (S/W). As the result of research in fish robot are belong to [1], [2], [3].

We are developing new contents that convert 2D fish picture to 3D hologram fish which is put in same aquarium leading by science and culture convergence research. In this paper, we have to find the fish robot objects in aquarium first. We try to find that object not use any external equipment only using software program (MATLAB) by obtained video frames.

We use the MATLAB program to calculate the image boundary detection easily by which is acquired the real-time video file. The mathematical algorithm is used not only a programming language, but applied in a programming environment as well. We can perform operations from the command line, as a sophisticated calculator, or we can create programs and functions that perform repetitive tasks, just as any other computer language. Also, it is reducing the time compare to any other program that is the reason why we select this program.

In this paper, we have studied for detecting the moving objects using boundary algorithm. We had set the camera to get video frames for fish robot in aquarium. Also, those fish robot have a different colors, it is possible to define the each other object using the proposed boundary algorithm.

In order to figure out the colored fish, we first use image processing that change RGB color to gray. Second, calculate the edge lined Canny's equations [4]. It indicates the optimal edge filtering and showed the filter function detects, which is find the edge with the minimum deviation from its true location. It changed the color to gray in accordance with distinguishing the objects and background. We can recognize the object that named this procedure is object detecting.

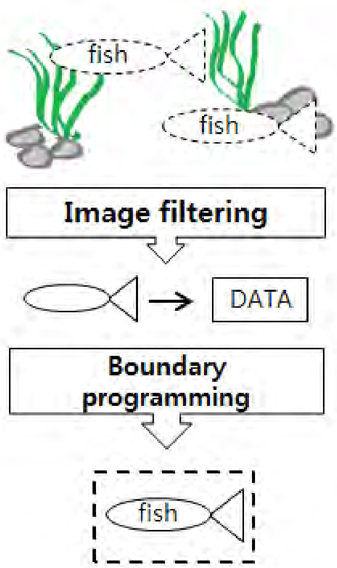
Finally, it applied to boundary algorithm to program we can find the fish robot creating boundary in their picture frames. Therefore, using boundary algorithm applied to MATLAB program we can find the fish robot or moving objects position not use any external objects only using the software programming. It suggests that we can identify the fish robot by indicating rectangular boundary surrounded the moving objects.

2. Modeling of Detecting Object

2.1 Modeling how to detect

We designed the modeling (figure.1) how to detect the fish robot in water tank. First of all, we captured the video by

using Sony camera. To load video into MATLAB by using 'VideoReader' command. After, we are applying the Image Filtering due to improve the pixels of images.
Second, we converted the captured images RGB to gray because of to find the objects and background easily. Third, we are applying the Canny's equations to calculate the object boundary and minimize objet noise ratio.
The final step of detecting moving fish is using boundary algorithm in programming.



(Figure1) The proposed detecting the fish robots

2.2 Boundary filter using mathematics

We also used the Canny's criteria for an optimal edge filter equations. We can calculate the filter function *f(x)* has maximal signal to noise ratio as shown in equation (1).

$$\mathrm{SNR} = \frac{\int_{-\infty}^{\infty} f(x)u(-x)dx}{n_0\sqrt{\int_{-\infty}^{\infty} f^2(x)dx}} \qquad (1)$$

where n_0 is the standard deviation of the additive noise. So, we try to maximize the quantity of object as shown in equation (2). Also, this equation showed that the filter $function(x)$ detects and edge with the minimum deviation from its true location if it is chosen in such a way that its first and second derivatives maximize the quantities.

$$\mathrm{L} \equiv \frac{\int_{-\infty}^{\infty} f''(x)u(-x)dx}{\sqrt{\int_{-\infty}^{\infty}[f'(x)]^2dx}} \qquad (2)$$

Finally, it showed that the output of the convolution of the signal with the filter $f(x)$ still contain minimal number of false responses if function $f(x)$ is chosen in such a way that its first and second derivatives maximize the quantity:

$$\mathrm{C} = \sqrt{\frac{\int_{-\infty}^{\infty}(f'(x))^2dx}{\int_{-\infty}^{\infty}(f''(x))^2dx}} \qquad (3)$$

P is represented performances measure, C is the maximized constant under the constraint C = Const. This is the way how appropriate for the detection of anti-symmetric features; i.e step or ramp edges when the noise in the signal is additive in equation (4).

$$\mathrm{P} \equiv (S \times L \times C)^2 = \frac{\left[\int_{-\infty}^{\infty} f(x)u(-x)dx\}\right]^2 \left[\int_{-\infty}^{\infty} f''(x)u(-x)dx\right]^2}{\int_{-\infty}^{\infty} f^2(x)dx \int_{-\infty}^{\infty}(f''(x))^2dx} \qquad (4)$$

3. Design of The Object Boundary Algorithm

3.1 Programming the boundary algorithm [6]

[Procedure 1] Initialize Foreground Detector

Rather than immediately processing the entire video, the example starts by obtaining an initial video frame in which the moving objects are segmented from the background. This helps to gradually introduce the steps used to process the video. The foreground detector requires a certain number of video frames in order to initialize the Gaussian mixture model. This example uses the first 50 frames to initialize three Gaussian modes in the mixture model. After the training, the detector begins to output more reliable segmentation results. The two figures below show one of the video frames and the foreground mask computed by the detector.

```
Foreground Detector =
vision.ForegroundDetector('NumGaussians', 3,
'NumTrainingFrames', 50);
fish =
vision.VideoFileReader('IMG_1155.avi');
for i = 1:150 frame = step(fish);
foreground = step(foregroundDetector, frame);
end
figure;
imshow(foreground);
title('Foreground');
```

[Procedure 2] Detect Object of Fish Robots

The foreground segmentation process is not perfect and often includes undesirable noise. The example uses morphological opening to remove the noise and to fill gaps in the detected objects. Next, we find bounding boxes of each connected component corresponding to a moving fish by using vision. We call this Blob Analysis object [5]. Blob Analysis is a fundamental technique of machine vision based on analysis of consistent image regions. As such it is a tool of choice for applications in which the objects being inspected are clearly discernible from the background. It used to calculate statistics for labeled regions in a binary image. The block returns quantities such as the centroid, bounding box, label matrix, and blob count. The Blob Analysis block supports input and output variable size signals. The object further filters the detected foreground by rejecting blobs which contain fewer than 150 pixels. The number of bounding boxes corresponds to the number of fishes found in the video frame. We display the number of found fishes in the upper left corner of the processed video frame.

```
se = strel('square', 3);
filteredForeground = imopen(foreground, se);
figure;
imshow(filteredForeground);
title('Clean Foreground');
blobAnalysis =
vision.BlobAnalysis('BoundingBoxOutputPort',
true,'AreaOutputPort', false,
'CentroidOutputPort',
false,'MinimumBlobArea', 150);
bbox = step(blobAnalysis, filteredForeground);
result = insertShape(frame, 'Rectangle', bbox,
'Color', 'green');
numfishes = size(bbox, 1);
result = insertText(result, [10 10], numfishes,
'BoxOpacity', 1,'FontSize', 14);
figure;
imshow(result); title('Detected fishes');
```

[Procedure 3] Reset Video Frames

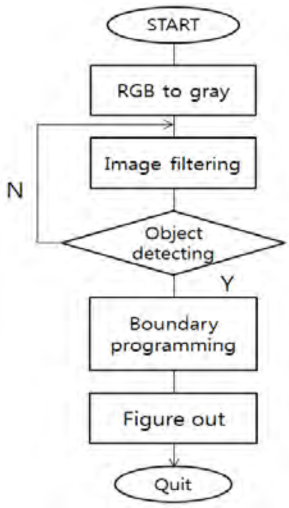
In the final step, we process the remaining video frames. The output video displays the bounding boxes around the fishes. It also displays the number of fishes in the upper left corner of the video.

```
videoPlayer = vision.VideoPlayer('Name',
'Detected fishes');
videoPlayer.Position(3:4) = [650,400];
se = strel('square', 3);
frame = step(fish);
foreground = step(foregroundDetector, frame);
filteredForeground = imopen(foreground, se);
bbox = step(blobAnalysis, filteredForeground);
result = insertShape(frame, 'Rectangle', bbox,
'Color', 'green');
numfishes = size(bbox, 1);
result = insertText(result, [10 10], numfishes,
'BoxOpacity', 1,
'FontSize', 14);
step(videoPlayer, result);
end release(fish);
```

3.2 Modeling the detecting object using boundary algorithm

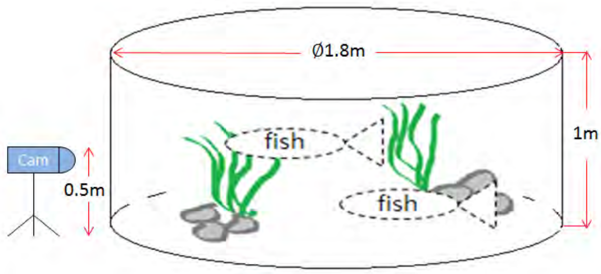
This is the modeling the detecting object using boundary algorithm. Video frames captured the moving objects is consisted of many frames of picture. So, before applying this algorithm capture the picture frames first. Then change the color and pixels by using RGB to gray and Image filtering to distinguish the objects and background. Also, it operates until finding the true object detecting.

After detecting the object, boundary detecting commands are operated. Finally, images that we want to figure out using this algorithm can be appeared.



(Figure2) Detecting the object boundary algorithm

4. Experimental Result



(Figure3) The detail of water tank and camera position

This figure is about the experimental condition of research. We designed the area water tank (height = 1m, diameter = Ø1.8m). To obtain the video capture frames, we set the water-proof Sony camera (height 0.5m). The running time of camera is three times of week, because depending on weather conditions disturbances are consist in water tank. The factor of disturbance is light which causes water reflections, thus it makes problems in captured frames to distinguish the objects and background. So, we try to capture different video frames and select the proper one to research it.

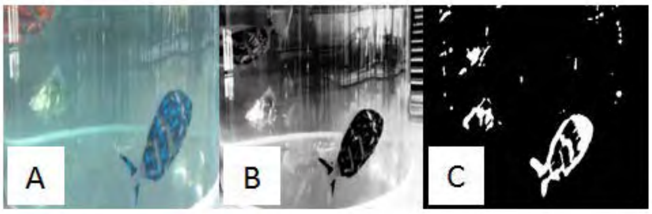
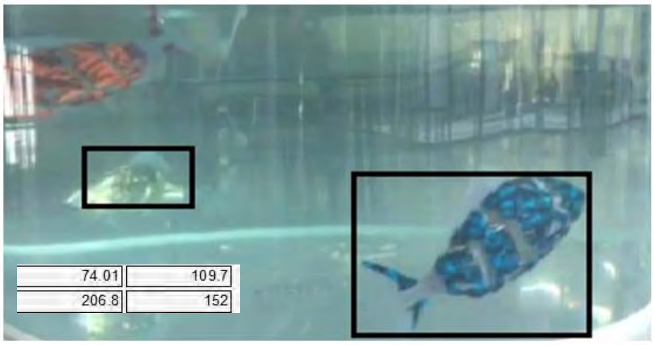


Figure4. The converted images using 'RGB to gray'
(A) Original (B) RGB to gray (C) Object boundary

These figures are the process of converted images using RGB to gray commands in programming. Here shown figures A, B, C. A figure is capturing by Sony camera. B and

C figures are after converted RGB to gray.



(Figure5) Detecting the fish by creating boundary to objects

This figure is the result of using boundary algorithm in program, which is applied in boundary algorithm equation show below. By using this formula, we can calculate and detect the boundary.

$$L' = \frac{\left(Z-(r-\frac{1}{2}\sigma^2)\right)^2}{2\sigma^2} + r \quad (5)$$

After through mathematical step, we operate the program that makes the boundary such as rectangular shapes and detects the moving fish robot.

5. Conclusion

This research conducts to detect the moving fish robot in aquarium using boundary algorithm. We designed the water tank, and set the Sony camera. Then we are modeling the step of research and use the MATLAB program to command. Also, we are applying the Canny's equations to identify the boundary by minimizing the noise. Finally, we use the proposed modeling and change that into programming commands. In conclusion, we can identify the position of moving fish robot through the phase of image processing and boundary algorithm.

Through this research we can design and program to detect moving object not using external equipment sensor.

Furthermore, we want to realization the new water world, which is 2D fish robot and 3D hologram put into same aquarium and convergence with together. We expect to use this research to detect fish robots and manipulate them and show the crowd fish robot futures.

REFERENCES

[1]Y. J. Kim, S. J. Kim, K. S. Yang, J. M. Lee, C. H. Yim, and D. H. Kim, "Design and Control of a Bio-mimetic Fish Robot," The Korea society of mechanical engineers, Vol.36, No.1, pp.1-7, Oct., 2012

[2]Zebra fish response to the robotic fish: preference experiments on isolated individuals and small shoals.G Polverino, N Abaid, V Kopman , S Macri and M Porfiri, Published in June 2012.

[3]Barron, J.L., D.J. Fleet, S.S. Beauchemin, and T.A.Burkitt. Performance of optical flow techniques. CVPR, 1992.

[4]Maria Petrou Panagiota Bosdogianni, "Image Processing The Fundamentals", p308-p311

[5]K. J. Shin, J. B. Lee, and Y. J. Seo, "Design of Autonomous Bio-mimetic Robotic Fish with Swimming Artificial Intelligence," The 2014 Fall Conference of the KIPS, pp.913-916, Nov., 2014.

[6]"Optical flow of Math works", [Internet], math works documentation, http://kr.mathworks.com