# A study of Detecting Fish Robot Position Using The Define Average Color Weight Algorithm

Amaranth Varma Angani*, Ju Hyun Lee**, Kyoo Jae Shin*
*Dept of ICT Creative Design, Graduate school, M.S. Course
**Dept of Digital Media Engineering,B.S.Course
Busan University of Foreign Studies Busan, Republic of Korea
e-mail : amaranth.angani@gmail.com, jh950616@naver.com, kyoojae@bufs.ac.kr

# 평균 색상 구분 알고리즘을 이용한 물고기 로봇 위치 검출 연구

아마르나 와르마 앙가니*, 이주현**, 신규재*
*부산외국어대학교일반대학원 ICT 창의융합학과
**부산외국어대학교 디지털미디어공학과
e-mail : amaranth.angani@gmail.com, jh950616@naver.com, kyoojae@bufs.ac.kr

## Abstract

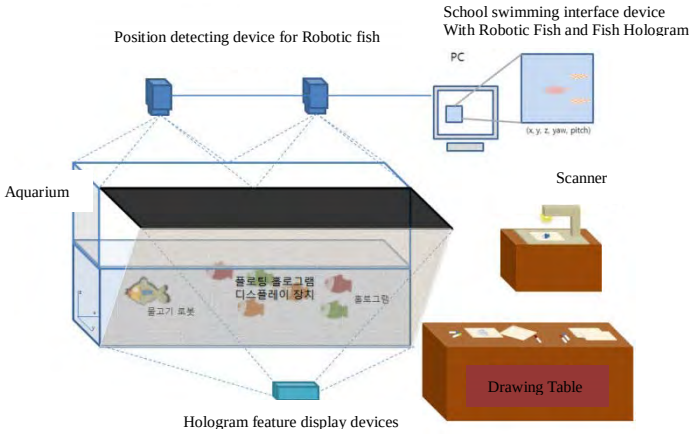
In this paper, the designed fish robot is researched and developed for aquarium underwater robot. This paper is a study on how the outside technology merely to find the location of fish robots without specific sensor or internal devices for these fish robot. The model of the fish is designed to detect the position of the optical flow of the Robotic Fish in the Simulink through Matlab. This paper intends to recognize the shape of the tank via a video device such as a camera or camcorder using an image processing technique to identify the location of the robotic fish. Here, we are applied to the image comparing algorithm by using the average color weight algorithm method. In this, position coordinate system is used to find the position coordinates of the fish to identify the position of the Robotic fish. It was verified by the performance test of design robot.

## 1. Introduction

Engineering design of Robots often inspired by the nature; Recent developed bio-inspired robots accurately have imitated various aspects of their live counter parts. Bio-inspired robotic dynamics is a fairly new sub-category of bio-inspired design. It is about learning concepts from nature and applying them to the design of real world engineered systems. More specifically, this field is about making robots that are inspired by biological systems [1]

We are developing a new culture, science and technology contents and in the aim of linking these all, we are using Robotic Fish and 3D Holograms through business of Scientific and Cultural Fusion Project. Our aim is to coincidentally present free swimming fish robots and 3D Holograms of fish and other sea creatures directly drawn by people in an Aquarium of Daejeon National Science Museum. In this paper, we used Matlab for the mathematical operations by detecting the Robotic fishes. To sum up, the robot is considered in this study seeks to analyze the motional behavior and detect the position of the Robotic Fish by using the Matlab for laboratory studies.

Presented method we have used for detecting Robotic fish is average color weight algorithm operated by Matlab. This designed robotic fishes for aquarium is tested to analyze the motion and flow of a specific fish or particular of schools ordered to swim to specific location. When this robotic fish passes a particular location value on a server, we communicate with the server, giving them an order to specific coordinates while this fish or a school swims free. To send a specific x, y coordinates to the fish, the size of the whole tank have to assign as coordinates. This thesis is about finding solution of a robotic fish position by using external technique without any particular sensor or inner equipment. At first, we have to recognize the form of fish to detect position of it by using only external technique. Detecting 3 axes fish robot position; we have used and discriminated to order by using average color weight algorithm. In this thesis, we are trying to find its position with the most little resource, using image process technique such as camera and camcorder including average color weight algorithm.



A hologram is a physical structure that diffracts light into an image. The term ‘hologram’ can refer to both the encoded material and the resulting image. A holographic image can be seen by looking into an illuminated holographic print or by shining a laser through a hologram and projecting the image onto a screen. Other methods of projecting and reflecting images are often described as holographic – or even misleadingly holograms, because they have an optical presence and spatial quality. [2]

We are planning to convert 2D Fish drawing which young spectator would drew and convert it into 3D Hologram using floating method similar to holographic display technology. Interacting with Hologram and Robotic fish, this will be the varied Digital Art Contents.

## 2. Modeling of Average Color Weight

Capturing video camera from aquarium, compare to other teams, “A Study of Detecting The Fish Robot Position Using The object Boundary Algorithm”, “A Study of Detecting Fish Robot Position Using The Compering The Image Data Algorithm”, who also have different idea of detecting fish robot, we implemented the work in only Matlab. We filtered the image converting video RGB to Gray by rotating the converting program for particular color, orange.

Here is some Matlab ordered document that we have reached to detect orange Robotic fish.

Step-1 Access Video with VideoReader

The *VideoReader* function constructs a multimedia reader object that can read video data from a multimedia file and also *VideoReader* for information on which formats are supported on your platform, the get method provides more information on the video such as its duration in seconds.

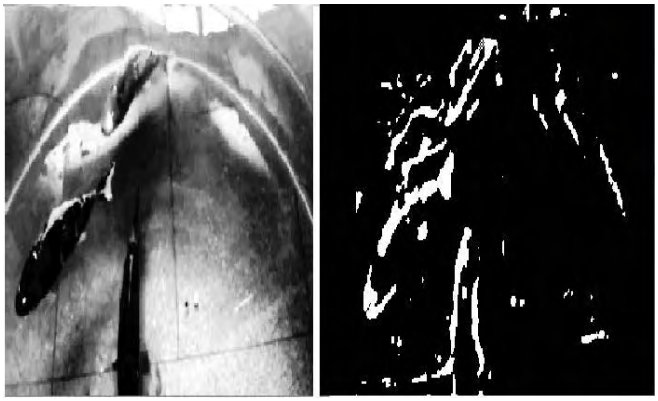
Use *VideoReader* to access the video and get basic information about it.

```
fish= VideoReader with properties:
General Properties:
                Name: 'IMG_1155.avi'
                Path: 'C:\Users\user\Desktop\imp'
            Duration: 22.7667
         CurrentTime: 0
                 Tag: ''
            UserData: []
  Video Properties:
               Width: 568
              Height: 320
           FrameRate: 30
        BitsPerPixel: 24
         VideoFormat: 'RGB24'
get(fish)
obj = VideoReader with properties:
General Properties:
                Name: 'IMG_1155.avi'
                Path: 'C:\Users\user\Desktop\imp'
            Duration: 22.7667
         CurrentTime: 0
                 Tag: ''
            UserData: []
    Video Properties:
               Width: 568
              Height: 320
           FrameRate: 30
        BitsPerPixel: 24
         VideoFormat: 'RGB24'
```

Step 2: Explore Video with IMPLAY

Use rgb2gray to convert the original video from RGB to grayscale. You can use the pixel region tool in implay to view pixel values. Specify the average pixel value (or a value slightly higher) as the threshold when you call imextendedmax. For this example, set the value to 50.The fish-tagging application processes the video one frame at a time in a loop.

```
video in implay.
obj=VideoReader('IMG_1155.avi')
orangefish=rgb2gray(read(obj,71));
orangefishValue=50;
noorangefish=imextendedmax(orangefish, orangefishValue);
sedisk = strel('disk',2);
noSmallStructures = imopen(noorangefish, sedisk);
subplot(131); imshow(orangefish)
subplot(132); imshow(noorangefish)
subplot(133); imshow(noSmallStructures)
```



(Figure 2) After Changing The image RGB to gray of Orange Fish Robot

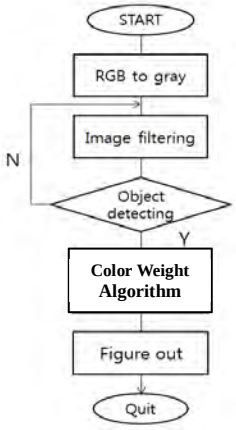
Step 3: Develop Your Algorithm

When working with video data, it can be helpful to select a representative frame from the video and develop your algorithm on that frame. Then, this algorithm can be applied to the processing of all the frames in the video.

For this fish-tagging application, examine a frame that includes deferent -colored like "Orange","Blue","Yellow" fishes. When an image has many structures, like deferent color video frames, it is useful to simplify the image as much as possible before trying to detect an object of interest. Typically, it takes a combination of techniques to remove these extraneous objects. Determine the average pixel value for these objects in the image. (Use rgb2gray to convert the original video from RGB to grayscale.) You can use the pixel region tool in implay to view pixel values. Specify the average pixel value (or a value slightly higher) as the threshold when you call imextendedmax. For this example, set the value to 50.The fish-tagging application processes the video one frame at a time in a loop. (Because a typical video contains a large number of frames, it would take a lot of memory to read and process all the frames at once.)

A small video (like the one in this example) could be processed at once, and there are many functions that provide this capability. Get the frame rate of the original video and use it to see tagged fish in implay.

```
nframes = get(obj, 'NumberOfFrames');
I = read(obj, 1);
taggedfishes = zeros([size(I,1) size(I,2) 3 nframes], class(I));
for k = 1 : nframes
singleFrame = read(obj, k);
I = rgb2gray(singleFrame);
noorangefishs=imextendedmax(I, orangefishValue);
 noSmallStructures = imopen(noorangefishs, sedisk);
noSmallStructures = bwareaopen(noSmallStructures, 150);
taggedfishes(:,:,:,k) = singleFrame;
 stats = regionprops(noSmallStructures, {'Centroid','Area'});
 if ~isempty([stats.Area]); areaArray=[stats.Area];[junk,idx] =
max(areaArray); c = stats(idx).Centroid; c = floor(fliplr(c));
 width = 2;
          row = c(1)-width:c(1)+width;
          col = c(2)-width:c(2)+width;
taggedfishes(row,col,1,k) = 255;
 taggedfishes(row,col,2,k) = 0;
taggedfishes(row,col,3,k) = 0;
end
end
frameRate = get(obj,'FrameRate');
implay(taggedfishes,frameRate);
```




(Figure 3) Object Detecting Using Color Weight Algorithm

## 3. The Proposed Average Color Weight Algorithm

### 3.1 Basic Idea of Image processing

**(1) RGB color space**

RGB color space or RGB color system, constructs all the colors from the combination of the Red, Green and Blue colors. The red, green and blue use 8 bits each, which have integer values from 0 to 255. This makes 256 * 256 * 256 = 16777216 possible colors.

RGB ≡ Red, Green, Blue

Each pixel in the LCD monitor displays colors this way, by combination of red, green and blue LEDs (light emitting diodes).When the red pixel is set to 0, the LED is turned off. When the red pixel is set to 255, the LED is turned fully on. Any value between them sets the LED to partial light emission.[3]

**(2) RGB color format & calculation**

RGB code has 24 bits format (bits 0..23):

| RED[7:0] | | | | | | | | GREEN[7:0] | | | | | | | | BLUE[7:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 23 | | | | | | | 16 | 15 | | | | | | | 8 | 7 | | | | | | | 0 |

```
RGB = (R*65536)+(G*256)+B , (when R is
RED, G is GREEN and B is BLUE)
```

**(3)Calculation examples of RGB Color**

**Yellow RGB Color**

```
Yellow RGB code = 255*65536+255*256+0 =
#FFFF00
```

**Blue RGB Color**

```
Blue RGB code = 0*65536+0*256+255 =
#0000FF
```

**Red RGB Color**

```
Red RGB code = 255*65536+0*256+0 =
#FF0000
```

**(4) Pixel**

The pixel (a word invented from "picture element") is the basic unit of programmable color on a computer display or in a computer image. Think of it as a logical - rather than a physical - unit. The physical size of a pixel depends on how you've set the resolution for the display screen.

The Pixel Information tool is a UI panel object, positioned in the lower-left corner of the figure. The tool contains the text string Pixel info: followed by the pixel information. Before you move the pointer over the image, the tool contains the default pixel information text string (X,Y) Pixel Value. Once you move the pointer over the image, the information displayed varies by image type, as shown in the following table. If you move the pointer off the image, the pixel information tool displays the default pixel information string for that image type.

### 3.2 Mathematical Analysis

**(1) Image Enhancement**

Image enhancement is the process by which we try to improve an image so that it looks subjectively better. We have used image enhancement to detect robotic fish. The pixels which represent different objects or parts of objects of image tend to have grey level values.

We used the mean and standard deviation of the distribution of pixels. In this way, we could enhance the variance inside each such window by using a transformation of the form:

$$g(x,y) = A[f(x,y) - m(x,y)] + m(x,y) \qquad (1)$$

$A$ is some scalar. To have their variance amplified most, chose the amplification factor A inversely proportional to $\sigma(x,y)$.

$$A = \frac{kM}{\sigma(x,y)} \qquad (2)$$

**(2) Karhunene-Loeve transformation: RGB to Gray**

The principal component analysis (*or Karhunene-Loeve transformation*) identifies a linear transformation of the coordinate system such that the three axes of the new coordinate system coincide with the directions of the three largest spreads of the point distribution.

To perform principal component analysis we must diagonalizable the covariance matrix of our data. The auto covariance function of the outputs of the assumed random experiment is:

$$C(i,j) = E\{(x_i(k,l) - x_{i0})(x_i(k,l) - x_{j0})\} \quad (3)$$

where $x_i(k,l)$ is the value of pixel $(k,l)$ at band, $i, x_{j0}$ is the mean of band $i, x_{j0}(k,l)$ is the value of the same pixel in band $j$, $x_{j0}$ is the mean of band $j$, and the expectation value is over all outcomes of the random experiment, i.e. over all pixels of the image:

$$C(i,j) = \frac{1}{N^2}\sum_{k=1}^{n}\sum(x_i(k,l) - x_{i0})(x_j(k,l) - x_{j0})\} \quad (4)$$

Since we have three bands, variables *i* and *j* take only three values to indicate R,G and B and the covariance matrix is a 3 × 3 matrix. For data that are uncorrelated, C is diagonal; i.e. $C(i,j) = 0$ for $i \neq j$. To achieve this we must transform our data using the transformation matrix A made up from the eigenvectors of the covariance matrix of the untransformed data. The process is as follows:

(1) Find the mean of the distribution of points in the color space, say point $(R_0, G_0, B_0)$.
(2) Subtract the mean grey level value from each corresponding band. This is equivalent to translating the RGB coordinate system to be cantered at the center of the pixel distribution.
(3) Find the autocorrelation matrix $C(i,j)$ of the initial distribution (where *i* and *j* take the values *R*, *G* and *B*).
(4) Find the eigenvalues of $C(i,j)$ and arrange them in decreasing order. Form the eigenvector matrix *A*, having the eigenvectors as rows.
(5) Transform the distribution using matrix A. Each triplet

$x = \begin{Bmatrix} R \\ G \\ B \end{Bmatrix}$ is transformed in to $y = \begin{Bmatrix} p_1 \\ p_2 \\ p_3 \end{Bmatrix}$ by: $y = Ax$; i. e. $y_k = \sum_i a_{ki}\, x_i$

This is a linear transformation. The new "colors' are linear combinations of the intensity values of the initial colors, arranged so that [4]

## 4. Experiment Result

Result of Detecting the robotic fish using color weight algorithm by using the camera, we captured the motion of fish to detect the position of the fish by color weight algorithm. These two crafted figures show each of pixel information and display the upper view of the position of the Robotic fish and beside the coordinates of robotic fish displays in arrow and column. In each of the pictures, the orange Robotic fish were detected with red dot as a detected mark.


(Figure 4) Former Experimental Analysis of Orange Fish Robot

Pixel info (A): Coordinate (156, 110)/ RGB [255 0 0]
Pixel info (B): Coordinate (210, 294)/ RGB [255 0 0]

Former figure (2) the swimming red fish was detected in the coordinate (156, 110). Later figure (3) coordinate of the red fish has changed to coordinate (210, 294). From this, we could see figure (2) orange Robotic fish swam from one position to another position detected by color weight algorithm.

## 5. Conclusion

As it had shown above, we detected the result of orange Robotic fish by using average color weight algorithm. However, there is still 25% predicted inaccuracy. For the better result, experiment and research have to focus more about exception in this method using Matlab with outside supplement. For example, the water tank, we used for this experiment, was too clear that sometimes it was giving wrong errors such as sunlight reflections, objects and people that are nearby. We expect we will have better accurate result after coloring the bottom of the tank to black avoiding few light reflection.

As Previously mentioned, our final result will be held in a Aquarium of Daejeon National Science Museum. This later version of this paper will be also considering about it's size, shape and depth. Probably more camera will be prepared considering all these conditions. Our experiment will non-stop until presenting z axe coordinate, 3 axes fish robot position, with more research.

Other than this, we have minor concern about the solution of imperceiving 3D holograms from camera that will be shot in the Aquarium.

## References

[1]https://en.wikipedia.org/wiki/Bio-inspired_robotics (Wikipedia)
[2]http://holocenter.org/what-is-holography/?gclid=CPrp-cyIjMgCFdcnvQod8V0Now, holography
[3] http://www.rapidtables.com/web/color/RGB_Color.htm, RGB Color
[4] Maria Petrou, Panagiota Bosdogianni,"Image Processing The Fundamentals, pp.125-138, July., 2005