Underwater Size Estimator

Group member: Jian Cao, Yi Ren, Ran Tian,
Chujian Bi, Jiawei Xu
Advisor: Kia Bazargan
Corporate Sponsor: Frank Dropps (DDR,LLC)
Date: 12.14.2014

Executive Summary

This project is going to design a prototype of underwater view system, which can estimate the size of underwater objects. The main purpose is to provide a better way for fishing. Hence, the product of this project can show the underwater view and estimate the size and the distance of the fish inside the view.

Two kinds of similar products are available for fishing. The first one is using the sonar detector system to locate the fish and detect the size of the fish. This one is mainly used for commercial fishing and already well developed. The second way is using a single underwater camera to show the underwater view, which is mainly used for scientific researches. The sonar detector can locate the fish and the single underwater camera can show the underwater view, but neither of them can tell the observer the exactly size and distance of the fish. To solve the demerits of these two products, a device, which is able to distinguish the fish and help the observer to estimate the size and the distance of the fish, is needed.

The product is designed based on the binocular system, which works like the human's eyes. Two underwater cameras are used to capture the underwater view at the same time and some algorithms are used to deal with the data. We use the software called Fish Phone on tablet to capture the video from camera through Wi-Fi and send the video to laptop through USB cable. The algorithms consist of the video read, edge detection, motion detection and size and depth calculation First, the motion detection can distinguish the fish from other still objects by subtracting images taken by the same camera at different time. Then, the edge detection determines the outline of all underwater objects. Finally, we use the size and depth calculation to get the size and the distance of the fish

Both data in the water and air have been collected and tested, and the errors of our algorithms are all acceptable, which are around 10%. The whole device works better when the water turbidity is better than 100NTU and the fish's distance from the cameras is less than 100cm. Overall by using dual camera system and some efficient algorithms on the laptop, the whole design can realize the underwater size estimating.

Table of Contents

Problem definition:

Background

Our project is aimed to design a device to help fishing. So far, two similar products are available for fishing. The first one is using the sonar detector system to locate the fish. This one is mainly used for commercial fishing and already well developed. It can generally distinguish the fish from other still underwater objects, but it cannot measure the exactly size of the fish and provide the underwater view. The second way is using a single underwater camera to show the underwater view, which is mainly used for scientific researches. The observer can see the underwater view and identify the species of the fish. However, it cannot detect the location and size of the fish. To solve the demerits of these two methods, the design of a device, which is able to show the underwater view and estimate the distance and size of the fish, is needed.

Keywords & Concepts:

<table>
<tr><th colspan="3">Key Words</th><th>Concepts</th></tr>
<tr><td rowspan="4">Underwater Size Estimation</td><td colspan="2">Wired Communication</td><td>Use RCA to DVD converter to connect the cameras and laptops to capture the video sent from the underwater cameras.</td></tr>
<tr><td rowspan="3">Binocular Algorithms</td><td>Edge Detection</td><td>Determine the outline of the objects</td></tr>
<tr><td>Motion Detection</td><td>Determine the moving objects</td></tr>
<tr><td>Size and Length Detection</td><td>Calculate the size and length of the moving objects</td></tr>
</table>

Customer Needs

- Required Features:
  - The device can distinguish the fish from other underwater objects.
  - The device can estimate the size of the fish shown on the screen and display the size estimate on the camera screen view.
  - The device has the operating depth of 30 feet underwater.
  - The device has the resolution of 1 inch at 20 feet.
- Optional Features:
  - The device can display the distance to the underwater object.
  - The device can display the size estimate on the camera screen view on real time.
  - The device has the resolution of 1/2 inch at 30 feet.
  - The device can match size estimation to the object on the camera screen view display.
  - The device can correct operation with more than two objects in the camera's view.
  - The device can adjust the displayed objects image on the screen view display to reflect its true size.

Product Design Specification

Parameter	Min.	Typ.	Max.	Description
Size (Camera Holder)	--	12"*12"*24"	--	Made by stainless steel and the size is required by the sponsor
Size (Camera)	--	12"*12"*12"	--	2 underwater cameras
Weight	--	0.9kg	0.95kg	2 cameras' weight & Camera holder
Working hours	6 hours	8 hours	10 hours	Rechargeable camera
Working depth	--	30 ft	--	

Working temperature	-10 °C	25 °C	40 °C	
Working platform	--	Above Android 2.0 & IOS 4.3	--	Smart phone
Estimation error	--	1 inch at 20 ft.	0.5 inch at 30 ft.	Different testing environment
Display resolution	--	240*320 Pixels	--	Determined by the camera
Video resolution	--	240*320 Pixels	--	Determined by the camera
Frame rate of the video	25fps	30fps	--	Determined by the camera
Compression Mode of the video	--	H.264	--	Determined by the camera

Block Diagram

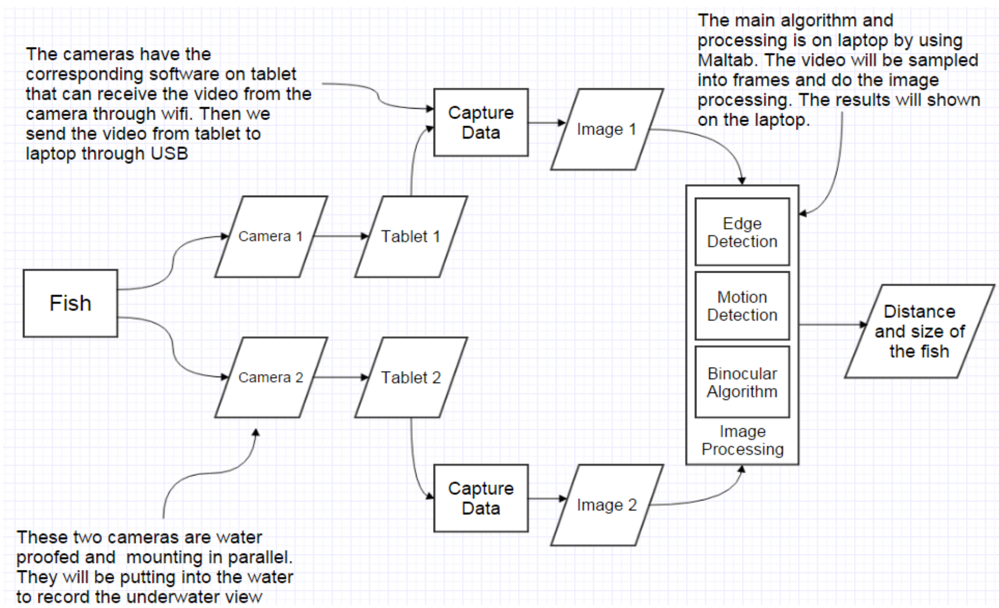


Figure 1: Block Diagram of the whole system

Design:

Summary of the design

This product aims to build an underwater size estimator that can see the underwater view and get the size and distance of the fish, which inside the camera's frame. The method is using dual camera and binocular algorithm to construct the 3D coordinate of the cameras' frame. In order to reduce the error and complexity of the algorithm, we mounting two cameras in parallel. We use the software called Fish Phone on tablet to record the video of two cameras and send videos to the laptop by using USB cable. Then, using Matlab to process the video. The processing includes sampling the frame of the video, doing motion detection, edge detection and applying the binocular algorithm to construct the 3D coordinate. Finally the size and distance of the object, which inside cameras' frame, can be obtained by the 3D coordinate. This is the overview of our design, the detail of each parts will be introduced in the next section.

Wired Communication:

Overview:

Since we need to estimate the size of fish under water, so we were provided with two underwater cameras. This camera is built with its own encoded WIFI hotspot and decoding software. However, the manufacturer refused to provide us the coding method of the software and the communication protocol between the camera and the cell phone. So we have to do reverse engineering to capture the data from the camera.

Approach:

In our plan there are three ways to do this: by capturing data from a simulated Android system; by capturing data wirelessly with Wireshark and decode the data package; by using wire to transmit the recorded videos to a laptop.

We tried three different simulation products to realize this function: Virtual Box, VMware and Blue stacks. They are the most popular software used for Android simulation. We run the Android virtual machine on the laptop and use Wireshark to capture the data that coming into the virtual machine. During our test, we found the Android virtual machine could only connect to WIFI signals occasionally and these two simulated Android system need to connect to different hotspots, which means it is impossible for one android simulator to receive 2 video packages at the same time. And it is hard for the software on the simulator to control the cameras, because commands are ignored for no reason. This should be caused by unstable WIFI connection. We repetitively press taking picture button on the simulator and use Wireshark to monitor the data. However, there are only 3 times that Wireshark shows that a jpg form of data is captured in 5 minutes.

For the second method, we use Scan function on Mac to find the frequency and bandwidth of the WIFI signal of those 2 cameras. They are at channel 6, 2.4GHz and 20MHz bandwidth. The security method is WPA2 personal type. We used Wireshark to run under monitor mode, which could capture all packages under a specific channel. This solve the problem that one laptop can't receive data from 2 cameras at the same time. After checking the data, we find most of them are about 1.5Mb and some of them are less than 1Mb. Each of them is consisted of a huge matrix, which is coded by 802.11 protocol and WPA2 security method. We search some information about this coding method on the Internet and try to do reverse engineering on those codes. But we failed, the code is too complex for us to decode.

Since we don’t have enough time to decode. We decided to use the simplest way that is to transmit the data by wire. We have two options in wired communication. First, we can capture the video by our phone with Fish phone App, then use wire to transmit two videos into the laptop and process them with Matlab. Second, we can use software and cables to read video stream from these 2 cameras, and store them in our laptop. We try both of these two methods and they works well.

We finally decide to demonstrate by using our phone. It’s easier for us to make sure that we can take videos at the same time. Since we didn’t reach our original goal that receiving and processing data at the same time. The communication part still needs a lot of improvement.

Binocular Algorithm

Overview:

The goal of our algorithm is to process the two videos captured from the two cameras and get the depth and size of the fish. The whole algorithm design is based on the math principle of 3D reconstruction on binocular images. 3D coordinates of the fish in the two images will be dealt with to calculate the size and depth of the fish. Our algorithm design can be divided into four parts – video read, motion detection, edge detection, size and depth calculation, Figure 2 shows us the overall diagram of algorithm.

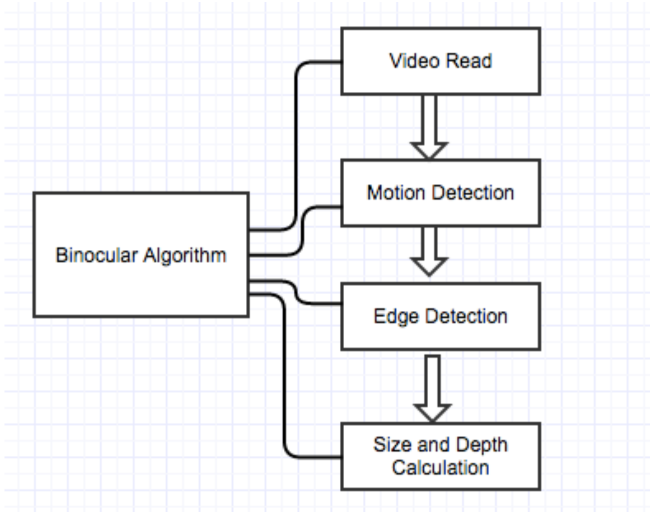


Figure 2: Overall Diagram of Binocular Algorithm: Video Read, Motion Detection, Edge Detection and Size and Depth Calculation

Mathematical Background:

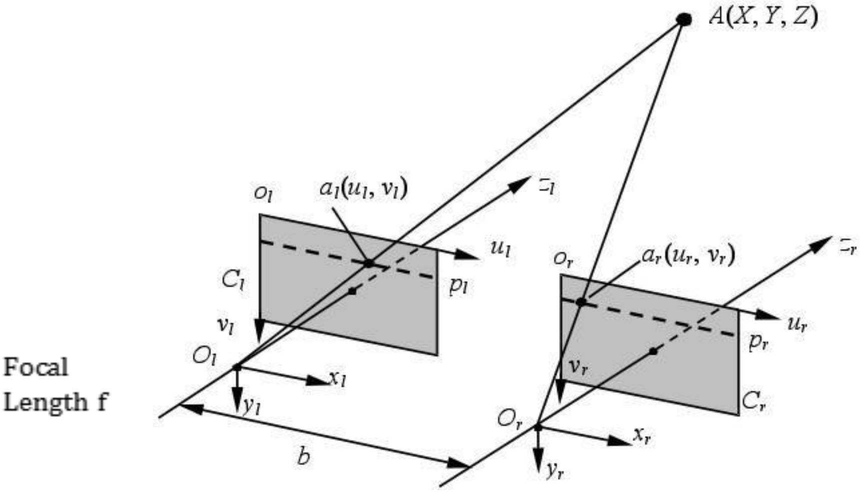


Figure 3: Mathematical Principle of 3D reconstruction[1]

Figure 3 shows how it works: b is distance between 2 parallel cameras. A can be any point in the space, and the projects of A into left camera and right camera are $a_l(u_l,v_l)$ and $a_r(u_r,v_r)$. The origin of the coordinate is the optic center of the camera, which is behind the image with the focal length f.

Assume A (X, Y, and Z) is the coordinate of the left origin point O_1. For the two points $a_l(u_l, v_l)$ and a_r (u_r, v_r), since the images of two cameras are in the same plane, $v_l = v_R = v$. According to the properties of triangle-similarity, we can get $u_l = \frac{X}{Z} \times f$, $u_r = \frac{|X-b|}{Z} \times f$ and $v_l = v_r = \frac{Y}{Z} \times f$. Here we define a parallax coefficient $d = u_l - u_r$, then we will find out

$$X = \frac{b \times u_l}{d}, Y = \frac{b \times v}{d}, Z = \frac{b \times f}{d}. \quad (1)$$

Same principle applies for the upper and lower images, we will get the same results except that parallax coefficient $d = v_u - v_l$.

Equation (1) gives us the spatial coordinates of any point A, we will use X and Z to decide the size and depth of the fish in our design.

❖ **Video Read:**

Goal: The videos captured from the upper and lower camera are the avi format, the goal of this sub algorithm is to extract the frames we want and save these frames as jpg format, which will be processed in the next parts.

Approach: The design is based on the VideoReader command in the Matlab. In order to get the desired frames in the videos, the method we use is to save the consecutive frames for a fixed interval. The frame rate for our camera is 30 frames/s, thus if the interval we set is 5 frames, the corresponding time interval is 5/30=0.17s. Imwrite is the main command to save the frame as jpg.

❖ **Motion Detection:**

Goal: The goal of this sub algorithm is to distinguish the fish from the other underwater objects. For a given image which contains different objects, we need to decide which one is the fish. There are two conditions for this part: the moving fish and the still fish.

Approach:

(1) For this sub algorithm, we regard fish as the only moving objects, then for two consecutive images from the saved images in video, we take the differences between these two. Then the result image will only contain the moving fish. If the fish move fast, the frame interval between consecutive images needs to be short in order to catch the fish. If the fish move slowly, then the frame interval should be long so that we can detect them. Figure 6 (Simulation Part) shows the image at frame 6 and frame 41 for the upper camera. Figure 7 shows the subtraction image between consecutive images.

We will need to apply this algorithm in the images of both lower and upper cameras, then we will deal with these images in the next parts.

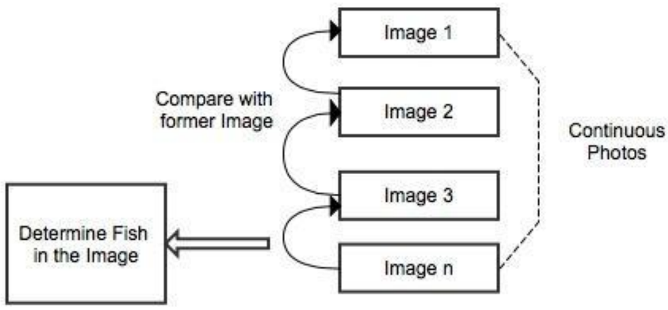


Figure 4: How to Recognize the Moving Fish

(2) However, if the fish is still, then this sub algorithm will not work. We will deal with this case in the Edge Detection part.

- **Edge Detection:**

Goal: For the images after the motion detection, we want to catch the "head" point and "tail" point of the fish in the two images. In other words, the goal of this sub algorithm is to extract the coordinate information of the two images.

Approach:

(1) For the moving fish, after motion detection, the image we get contains the fish and some noises. Firstly we use the Sobel Edge Detection [2] to extract the edge points of the images.

Then we need to eliminate the noises in the images. The way we use to filter the noise is Gaussian Filter [3], which essentially convolve this binary image with a certain matrix. In other words, the image will be smoothed after Gaussian Filter. Figure 8 shows us the result before and after using Gaussian Filter.

To get the "head" point and "tail" point of the fish, we use a rectangular to trunk the fish in the binary image, thus the rectangular vertices can be approximately regarded as the "head" point and "tail" point. To realize the rectangular trunk, we use the connected components and regionprops function in the Matlab. After marking the connecting components [4] in the binary image, the regionprops [5] command will give us the properties of these connected components including the rectangular trunk. Figure 8 shows us the results after we trunk the fish.

(2) For the case of still fish, the motion detection can't detect the fish and we can only deal with images containing other objects. In our design, we are able to determine the fish that are relatively larger than other objects. The main algorithm is almost the same as above, the difference is that now there are many rectangular trunks in the image instead of one (see Figure 11), the reason for that is we also trunk other objects in the image. To distinguish the fish from other objects, we choose the one with maximal area among all rectangular trunks. This is corresponding to our assumption at the beginning that other objects (including noise) are relatively smaller than the fish. Figure 12 shows us the result of final trunk.

❖ **Size and Depth Calculation:**

Goal: To determine the size and depth of the fish using the binary image with rectangular trunk.

Approach: Given the equation (1) $X = \frac{b \times u_l}{d}$, $Z = \frac{b \times f}{d}$,

The 3D coordinate of "head" point is $X = \frac{b \times u_{head}}{d}$, $Z = \frac{b \times f}{d}$

The 3D coordinate of "tail" point is $X = \frac{b \times u_{tail}}{d}$, $Z = \frac{b \times f}{d}$,

Then the length of the fish $L = \frac{b \times (u_{head} - u_{tail})}{d} = \frac{b \times u}{d}$

The depth of fish $D = Z = \frac{b \times f}{d}$.

Notice d=$v_{upper} - v_{lower}$=Δv*m, u=$\Delta u * m$, where Δv is the parallax coefficient (in pixels), Δu is the length of rectangular trunk (in pixels), and m is the physical size of one pixel.

Finally we get: $L = \frac{b \times u}{d} = \frac{b \times \Delta u}{\Delta v}$; $D = \frac{b \times f}{\Delta v * m}$;

Δv and Δu can be determined by the rectangular trunks in the lower and upper images, thus L will be calculated. To get a more accurate estimation on size, we need to adjust the value of b for different distances. See more details in the second advice in the User Manual in Appendix.

To calculate the depth, we need to know the value of $\frac{f}{m}$, f and m are two parameters of the camera, but these are unknown for us and it changes with distance. Thus we do a method called reverse engineering to find the $\frac{f}{m}$ value, which means we will compute $\frac{f}{m}$ for a given distance. Figure 4 show us the look-up table we measured at b =0.12m;

Distance(cm)	20	30	40	50	60	70	80	90	100
$\frac{f}{m}$	268	257	245	235	223	215	207	200	195

Table 1: Look-up Table in Underwater for b=0.12m

In order to roughly estimate the depth of the fish, we choose $\frac{f}{m} = 230$. However, this will give us big errors when the distance is greater than 1m (see Table 5).

Mounting

Goal: To get more accurate results, we need the two cameras are parallel to each other and both of them need to be parallel to the camera holder's base. Two cameras also need to be fixed well to prevent the water flow's influence.

Approach: The figure 5 shows the mounting system for our device. The whole mounting system consists of four cuboid stainless steel rods, five L brackets, two holders for cameras and several screws. The L brackets are used to fix the rods and the two cameras are fixed by four special screws. Based on the testing errors, we think this mounting system is good enough for the whole device.

Figure 5. The Mounting System

Design Evaluation

Evaluation by prototyping

The sample we take is from the swimming pool in the Rec center. The simulation figures are listed below. For the upper cameras for moving fish, the simulation for the lower camera is the same as upper one. b=20cm, D=90cm, and L=34cm:

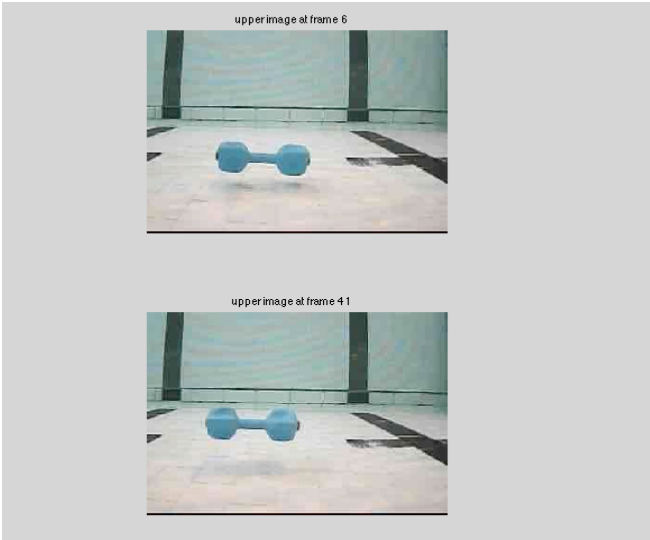


Figure 6**:** The Upper Image for Two Consecutive Frames

Figure 7: Subtraction Image of these two Consecutive images for Upper image

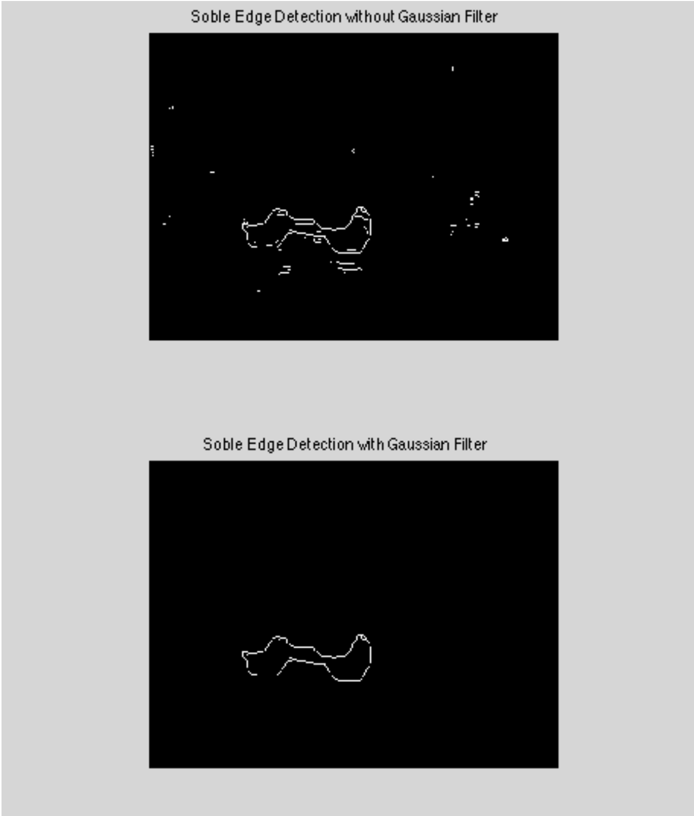


Figure 8: Sobel Edge Detection with & without Gaussian Filter for Moving Fish

Figure 9 Add Rectangular Trunk for Moving Fish

The following figures are the simulation results for still fish, b=20cm, D=76cm, and L=45cm:

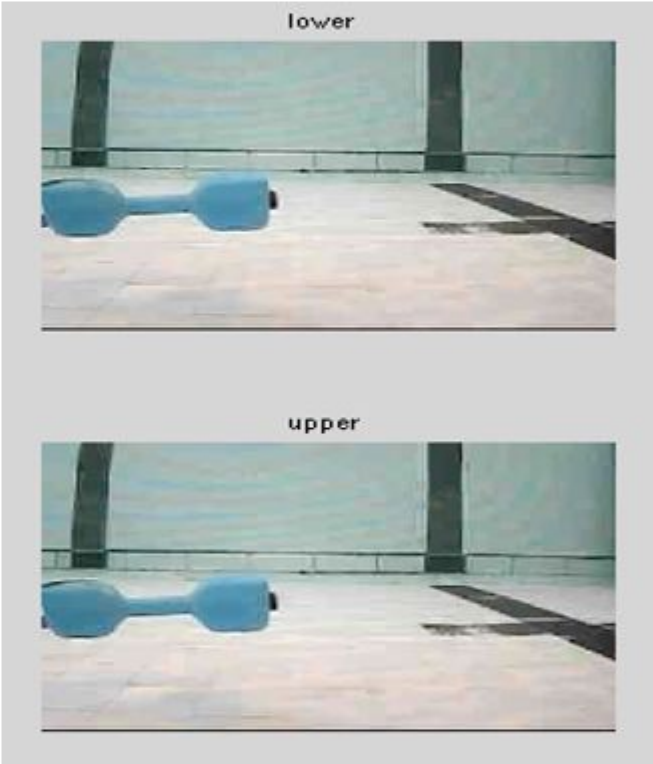


Figure 10 Upper and Lower Image for Still Fish

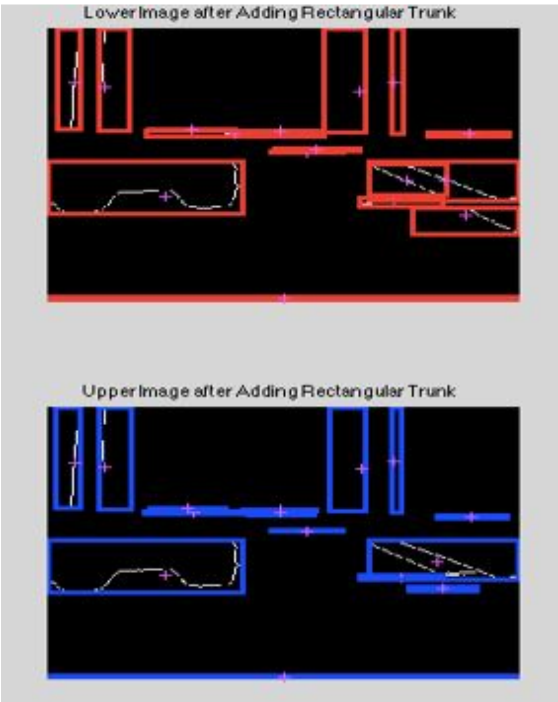


Figure 11 Upper & Lower Image after Adding

Rectangular Trunk

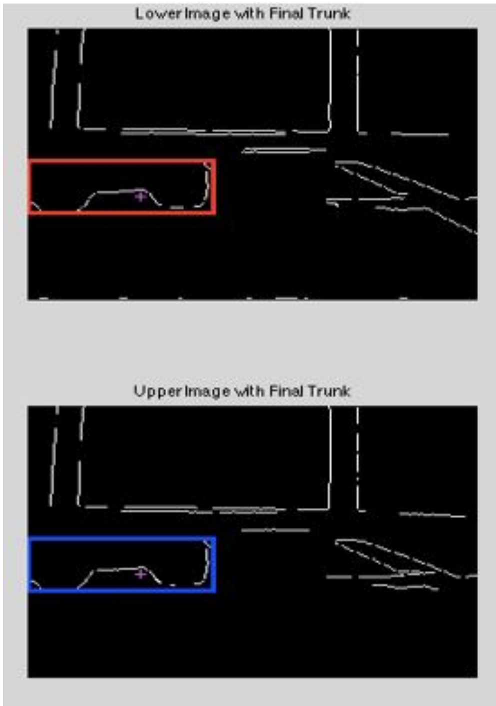


Figure 12 Upper & Lower Image with Final

Rectangular Trunk

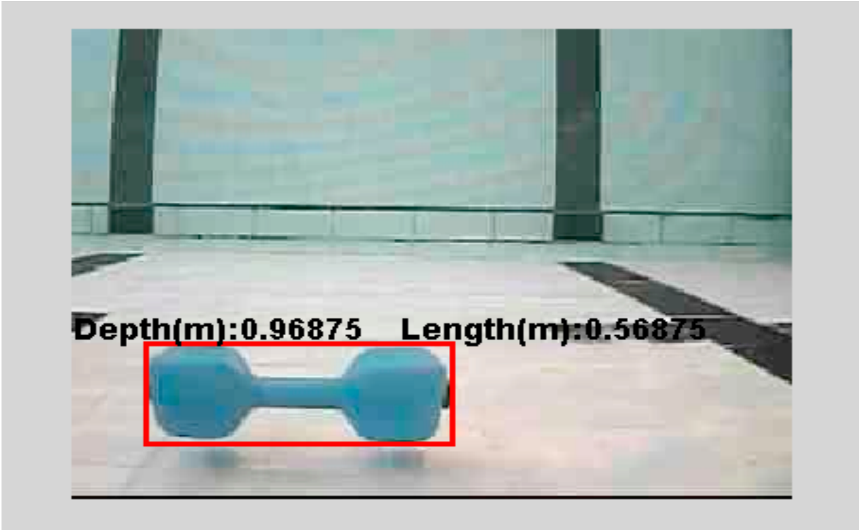


Figure 13 Final Results after our Algorithm

Evaluation by analysis

Here are testing results in the underwater for size estimation with b=12cm, we test it in the bathtub, due to the length limitation of the bathtub, the longest distance we can measure is 80cm.

Distance(cm)	20	30	40	50	60	70	80
Actual length(cm)	14	14	14	14	14	14	14
Estimated Length(cm)	11.3	15.4	13.7	12.1	15.5	15.6	15.9
Estimated Error (%)	19.2	10	2.14	12.8	10.7	11.4	13.5

Table 2 Size Estimation in the Bathtub with b=12cm

Distance(cm)	20	30	40	50	60	70	80
Estimated Depth(cm)	13	25	37	48	74	80	93
Estimated Error (%)	35	16.7	7.5	4	10	14.2	16.25

Table 3 Depth Estimation in the Bathtub with b=12cm

Here are the testing results we test it in the swimming pool in the Recreation Center. Since they only permit us testing in an hour, we only test the case with b=12cm.

Distance(cm)	91	107	137	167	198	228	260
Actual length(cm)	45	45	45	45	45	45	45
Estimated Length(cm)	56	57	60	67	62	70	72
Estimated Error (%)	24.4	26.6	33.3	48	37.8	55.6	57.7

Table 4: Size Estimation in the Swimming Pool with b=12cm

Distance(cm)	91	107	137	167	198	228	260
Estimated Depth(cm)	110	130	172	220	297	420	503
Estimated Error (%)	19	21.4	25.5	36	50.5	84.2	93.4

Table 5: Depth Estimation in the Swimming Pool with b=12cm

Error Analysis: Comparing Table 4 & 5 with Table 2 & 3, the estimation errors are larger when the distance is greater than 1m. This big error is caused by the low resolution of the camera and small distance (b) between two cameras. To decrease the error estimation, we need to choose two cameras with high resolution and increase b.

Table 6 shows us the results of two iPhone with b=12cm. Comparing with Table 4 & 5, the size and depth estimation error is much smaller.

Actual Depth(cm)	50	80	100	120	150	200
Estimated Depth(cm)	47.9	79.1	102.7	117.4	138.2	184.6
Estimated Size(cm)	20.50	20.82	21.07	21.23	21.45	21.94
Size Estimation Error	2.5%	4.1%	5.35%	6.15%	7.25%	9.7%
Depth Estimation Error	4.2%	1.8%	5.4%	5.2%	7.8%	7.7%

Table 6: Testing Results with Two iPhone in the Air with b=12cm

Table 7: shows us the results of two iPhone with b=30cm.

Actual Depth(cm)	50	80	100	120	150	200
Estimated Depth(cm)	48.4	79.3	100.01	118.4	158	214.2
Estimated Size(cm)	20.29	20.52	20.61	20.58	20.96	21.3
Size Estimation Error	1.5%	2.6%	3%	2.9%	4.8%	6.5%
Depth Estimation Error	3.2%	0.9%	0.1%	1.8%	5.3%	7.1%

Table 7: Testing Results with Two iPhone in the Air with b=30cm

Comparing Table 6 with Table 7, the size and depth estimation error with b=30 is smaller than the case that b=12cm.

Figure 13&14 gives us the size and depth error tendency for our camera and iPhone.

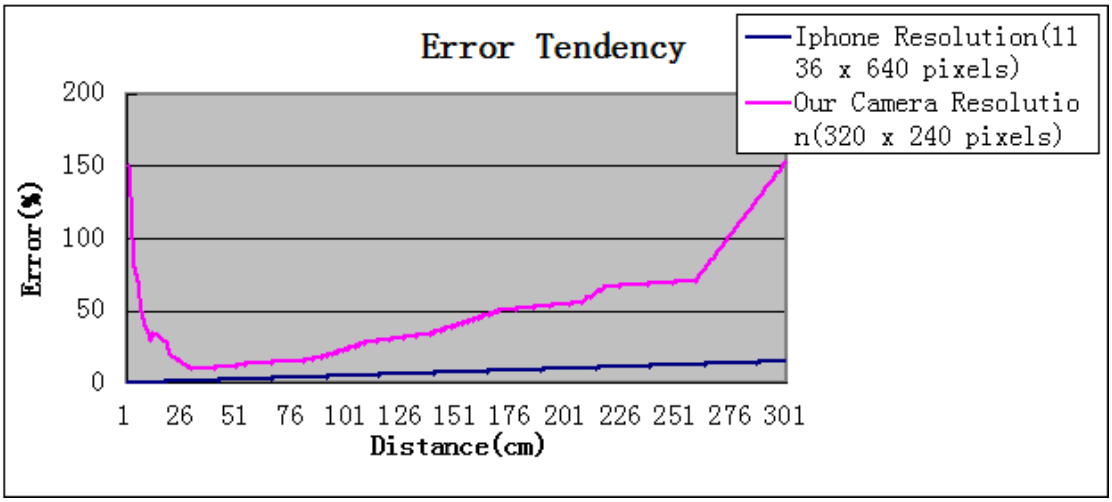


Figure 13 Size Estimation Error for Our Camera and iPhone

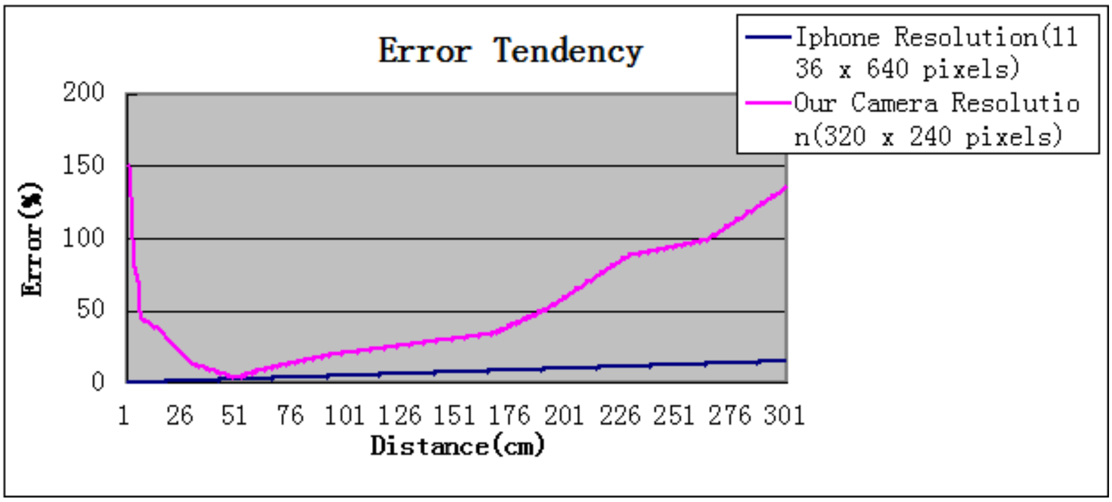


Figure 14 Depth Estimation Error for Our Camera and iPhone

Evaluation by customer

We interviewed some customers and most of them think this product is pretty interesting. They think our product does meet their expectation that knowing the size and distance of the object they are trying to observe. However, the algorithm will consider all the moving objects as fish, and the product cannot works well if the fish is not moving. It would be wonderful if Matlab could work simultaneously when the camera is recording.

Conclusion & Recommendations:

Specification Discussion:

Our design meets the product design specification except the estimation error. Due to the low resolution of our cameras, our design can only keep the error smaller than 15% when the fish is less than 1m away from the cameras. If the fish is too far away from the cameras, we found some edges of the fish will lost in the video, which will increase the estimation error.

We did not meet too much cost and working environment (for the device) issues, because all materials are provided by the company, which already met the specification. For the size of the whole device, which is completely determined by the size of the cameras and the mounting systems. We found several designs for the mounting system and the sponsor also provided us with a design, but it was too hard for EE students to fabricate a fancy mounting system. So we just decided to use the one mentioned above to meet the specification.

Compared with the similar products in the market, our product can help people to locate the fish and also estimate the size of the fish. Also our product is relatively cheaper and easy to build. Meanwhile our device is not limited to the speed of the fish, because we can change the frame interval between consecutive images for different fish. Meanwhile, if the fish is less than 1m away from the cameras, out testing error can be smaller 15%, which is fair enough for people to realize how big the fish they are looking at is. However, our product is mainly designed for fishermen, so our product's working distance is too small. For now, we only can detect the still fish when it is larger than anything else and we only test our device with small noise.

Future Improvement:

So for the future improvement, firstly we need use two cameras with high resolution. The table below show the testing results from two iPhone 5.

Actual Depth(cm)	50	80	100	120	150	200
Estimated Depth(cm)	48.6	79.1	97.8	117.4	148.2	197.6
Estimated Size(cm)	20.21	20.30	20.01	20.00	20.80	21.40
Size Estimation Error	1.05%	1.5%	0.3%	0	2.4%	4.2%
Depth Estimation Error	2.8%	1.8%	4.4%	5.2%	3.6%	4.8%

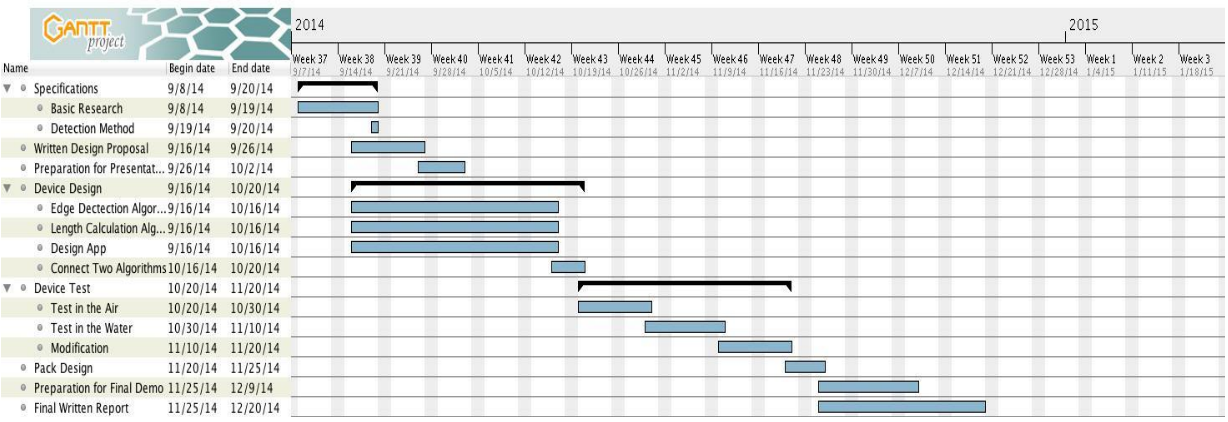
Table 8: Testing results of two iPhone 5 (much better than our cameras)

As we can see, by using high resolution cameras, we can get more accurate results and can calculate the size of objects far more than 1 meter away. We also need to design better algorithm for the still fish.

Currently we use the Matlab do all the calculation, which means the fishermen need to take a laptop when they are going to fish. So if we can design a phone or tablet APP, which applies all the functions then we can make the whole device much smaller and solve the wireless communication issue. And if the future device can estimate the size of the fish in real time, it will be much popular in the market.

Reflections on the design process:

- **Timeline**



When we got the cameras from the sponsor, we found the cameras are fish-eye, which means we cannot test it in the air. So we delete the part of testing in the air and give more time for the testing of our algorithms. Meanwhile, after the discussion with our advisor, we decided give up designing the APP. Except these two things, we almost follow the timeline above.

❖ **Budget**

Material	Mode	Number	Price
Waterproof Camera	SONY H.264	2	$600
Connection Cables	Video Transmission	2	$50
L brackets	--	5	$10
Steel Rod	--	3	$10
Screw	--	10	--
Total	--	--	$670

This product is sponsored by the company, so all the materials are provided by the company. The cameras are the most expensive one in our project and rest things are mostly used for the mounting system.

Comment on design process

When the first time we know details about this project, we almost gave up at first because we know nothing about this image processing, java and wireless communication. But after several months, we were encouraged by our advisor and we learnt something new by ourselves, which gave us some valuable experience cannot got from classes. Sometime we argued about some certain parts about the design and doubted about other group members' choice, but finally we meet the requirements of the company. So we think our design is very successful.

Individual work

Yi Ren

At the beginning of the project, we considered about showing size and distance of the fish on the tablet and decided to use Bluetooth to achieve the interface between tablet and laptop. Hence, I was responsible for the Bluetooth communication between tablet and laptop. Since we will use Matlab to do the image processing, I focused on how Matlab could interface with other device through Bluetooth. I found Matlab has the Bluetooth interface in the instrument control toolbox that can call laptop's Bluetooth adaptor to achieve the Bluetooth Communication with other devices like smart phone. I could send the image from an Android phone to the Matlab directly through Bluetooth and I could send text from Matlab to that phone. I also found some Java codes about calling Bluetooth in the application of Android device. I hope to make an application to show the results on tablet but I failed to achieve that goal. Because no one in our group is familiar with Java and we need more time to focus on other parts of the project. Finally, we decided to show our results only on the laptop.

Then, I worked on capturing the data from camera with Jiawei Xu. Our camera has corresponding application on tablet to receive the data from camera through wifi. Hence, I tried to use Android emulator to run that application on laptop. I tried the emulator in the Android Software Development Kit but I found the emulator cannot emulate the wifi system. I also tried several Android emulators but none of them could emulate the wifi system. Then I tried the wire connection between camera and laptop. Our camera has a RCA output cable and I used a RCA to USB converter to capture the video from the camera. That is one of our method to capture the data.

I designed the poster and joined the underwater test in the swimming pool of Rec Center.

Ran Tian

For this project, I am mainly responsible for the algorithm part. I was arranged to work on the algorithm at the first several meetings. I searched possible papers related to the project at first and then worked on these papers. First I study the math model of the 3D reconstruction which is the fundamental theorem for our project. Then I determine the main structure of our algorithm (binocular algorithm)- video read, motion detection, edge detection, size and depth calculation. And then I am fully taking part in constructing and debugging these sub algorithms. I come out the main idea of extracting the fish in the image- using rectangular box to trunk the fish. And for the edge detection, I find the concepts of connected components which are helpful in noise filtering. And I also study the relevant papers about calibration and decide to use the “reverse-engineering” way to determine the value f/m. These are the detailed information about my work. In all, my main contribute to our project is succeeding the algorithm part.

Besides, I am taking part in testing our algorithm in the bathtub and in the swimming pool. And I am working on debugging the codes on these samples data. Then I prepared for the final demo using the sample videos. Also, I am working with sample data with two Iphones and comparing them with our cameras. Thus I conclude the idea that choosing two cameras with high resolution in the Future Improvement.

For the report and presentation, I am also mainly responsible for the algorithm part. And I also write the simulation and test parts. Besides, I give some advice for our final poster. Then I am also responsible for modifying the final algorithms (motion algorithm and still algorithm) in our report.

Jian Cao

I am the team leader of this project, so the first thing I am responsible for is the contact between the advisor, sponsor and group members. I am also for the editing of paper work. Then for the design part, I am mainly responsible for the motion detection and the mounting system. All the team members participated in the testing of the design.

This is the first time for me to be a leader of a research project, so everything is new to me. When every task is given, I need to consider how to divided it evenly and then inform the advisor and the sponsor as soon as possible. All of my group members are all Chinese, so the paper work is painful for us. The grammar and some logical problems are unavoidable, so I need to make appointments with the writing center and contact with the TA to make our paper work better.

For the mounting system, firstly I consult an advisor from the ME shop to design the mounting system for the project. However when I started to fabricate it in the ME student shop, I found it was too hard for an EE student to use the equipment efficiently. So I discussed with the sponsor and the ME advisor, we decided to use some stainless steel rods, L brackets and screw to make a simple one. After the sponsor gave me the material, Jiawei Xu and I spent half week in the ME student shop fabricating the mounting system.

For the motion detection, the idea is actually coming from our discussion with our advisor and sponsor. All the group members are not familiar with the image processing, so I searched online and asked some professors about some optional designs. Based on what I learnt in EE4505, which is relevant to image processing, I decided to use the method that subtracting two pictures taken by one camera at different time. However this method cannot detect the still fish, so I discussed with the Ran Tian to use his edge detection algorithm to detect the still fish.

We tested both in the bathtub and the swimming pool. We used some toys and rules as the fish and tested with different distance. All the group members participated in the debugging of the algorithms to minimize the error.

Jiawei Xu

In our project, I am mainly in charge of the communication part. Actually at first, we didn't realize communication was a problem until we got the news that the manufacturer refused to provide us information about the communication protocol between cameras and Apps. Without the code of Fishphone Apps, the only way to realize the function of processing the video stream simultaneously is to do reverse engineering on the camera. Since then, Yi Ren and I started working on this problem.

Since none of us has ever studied even a little about network. So I searched a lot of information about the way of communication between two devices. When we have an overview of what is reverse engineering in wireless connection. We went to CSE IT office to collect some more detail and practical suggestions, such as what software we should use. Finally we decide to try three different ways, they are capturing data from a simulated Android system, capturing data wirelessly with Wireshark and decode the data package and the last option is using wire to transmit the recorded videos to a laptop. I simulated in VMware and Virtual Box, there is no wireless connection option in the Android virtual machine. So we tried the second way, I ran my Mac under sniffer mode, and found out the frequency and bandwidth of the wireless signal of the camera. Then I captured the entire data package under that specific frequency and read it with Wireshark. But it is encoded with WPA2 security method, and we failed to decode it. At last since the system is coming to end, we have to abandon the thought of processing the data simultaneously. We decided to use our last option to get the video by wired connection.

I also helped Jian to make the mounting system by cutting steel rods and drilling holes into it. I also joined the testing part by diving into the swimming pool to place our product and the object to be tested.

In conclusion, I mainly worked on the wireless connection part, especially decoding part. I searched a lot of information about 802.11-protocol and WPA2 security method. However, I am novice, decoding WPA2 data package in 1 month is too hard for me. At last I failed, but I learned a lot about intercepting and cracking.

Chujian Bi

In our project, I mainly focus on the binocular algorithm. At the very beginning, our project needs a specific method to realize our goal. I read some research papers to figure out which way is better for us. I found out the binocular algorithm using dual-cameras would be easy to realize and design, then I sent some articles and information to our group members to help us understand the fundamental idea of binocular. Moreover, I contacted with some professors who are experts in the field of image processing to make sure the procedures of this algorithm, namely camera calibration, stereo rectification and stereo matching. After that, I did the single camera calibration in and stereo calibration in Matlab and successfully got the inner and extrinsic parameters, including cameras focal length and three dimensional re-constructional coordinates. I found out some inner parameters like rotation and transmission matrix can be simplified because our mounting system can mechanically fix two cameras parallel instead of putting them in the different directions randomly. Therefore, we tried to build up a look-up table to simplify the design.

Then I collaborated with Ran tried to figure out the stereo rectification, we did some experiments to calculated the f/m(a parameter in the formula) in the air and water to build up the look-up table to be used in our algorithm. I helped him to calculate the look-up table and debug the code. Also, I tried to improve the accuracy by using bundle adjustment to get the physical pixel size. What is more, after the first test in the bathtub, I realized we need a larger experimental environment to move further. I contacted with staff at recreation center to rent their swimming pool for tests. At the end of our project, I finished the error analysis, compared the detection result of our cameras and iphone 5 to draw the conclusion our error mainly decided by the low resolution of the cameras and make two graphs to show the relationship and tendency of error between our camera and iphone 5.

I joined every test and discussion. For the presentation parts, I mainly introduced the limitations and future improvements.

References

[1] http ://image.baidu.com/i?,ct=503316480&z=0&tn=baiduimagedetail&ipn=d&word

[2] http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm

[3] http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

[4] http://en.wikipedia.org/wiki/Connected-component_labeling

[5] http://www.mathworks.com/help/images/ref/regionprops.html

Appendix: User Manual:

- For the moving fish, use the "motion_algorithm",for the still fish, use the "still_algorithm".
- To get a more accurate estimation, adjust positions of two cameras to change value b. For distance d is less than 1m, suggested value for b is 0.12m. For d is greater than 1m, the suggested value for b is 0.3m-0.4m, but customers need to make sure the fish show in both videos of two cameras.
- For the moving fish, if the fish moves fast, decrease the value of frameinterval, if the fish moves slowly, increase the value of frameinterval. Continue adjusting this value till the rectangular box trunk the fish.