Osaka University -OUXT Polaris-

Journal Paper about the development for the 2016 RobotX Competition

Wada Suisei, Kataoka Masaya, and Kimura Riku

***Abstract*–This paper is written as the journal paper for the 2016 Maritime RobotX Challenge in Hawaii.This paper consists of mainly five parts: 1.Introduction, 2.Design Strategy, 3.Vehicle Design, 4.Experimental Results, 5,References.**

1 Introduction

Nowadays, many of researchers are interested in autonomous vehicles around the world.That is because, most of the vehicles in the world are controlled by human, but they cannot avoid not to make mistakes and sometimes fall in dangerous situation or meet in an accident unfortunately.So researchers are considering that the autonomous vehicle controll system may be one of the solutions to avoid traffic accidents from human error.

Ocean traffic have also experienced many severe accidents for long time, and some of them were caused by human error:e.g. TITANIC.Ocean vehicle is more unstable than grand one and there is no road.So it is more difficult to control and detect obstacles.We sure that autonomous systems or some control assisting systems included some sensors and algorithms, will contribute to avoid severe accidents on the ocean.We think this competition is very significant for proceeding these technologies,so we deceided to participate to this Challenge.

2 Design Strategy

In this competition, mainly 8 tasks are set.So in the beginning of team activity, we discussed how functions will be needed to clear the tasks.In the discussion, we used the well-konwn sequence that human execute to control vehicle.The sequence consists of three steps: 1.Recognition, 2.Judgement, 3.Act. We applied them to autonomous robots and divided required function as next.

1.Sensors
This part provide the information of the environment. It is equivalent to Recognition Function.

2.Computer and Algorithm
This part is core components of autonomous vehicle. It decide the direction of the robot's movement.Compute environment information and judge next behavior.It is equivalent to Judgement Function.

3.Hardware
This part generate action to the outside of the robots like motors.And for generate forces to the environment,it is included some power equipments.Also, the robots have to be communicate to out, so this part is also included some communication parts.

In the beginning of the discussion, we considered how functions are required for the robot in order to clear the tasks.Before designing sensors,we thoght that abstract function list is essential.So we made Table1 and classify the abstract function.Next we show the sequence to decide vehicle design and strategies for each function above.

2.1 Sensors

This time, few memebers are participated in our team, and our budget cannot afford to develop underwater system. So we give up to participate in the underwater tasks.So, accorgind to the Table1,the robots need sensors to get next information: "Shape","Color","Obstacle Relative Position","Sound source Detection","GPS."We discussed how to get these information.We had to develop with lower cost, so we will use usb-camera for detecting shape and color.And waterproof microphone will be used for detecting sound source under the water.
For obstacle detection, we considered at first to use 3D Laser Range Finder, but this sensors have a problem to adopt.LRF is very expensive sensors.One of our sponsors rent LRF(Fig.1), but the range is not so long(less than 30 [m]), but we estimated more than

Table 1: Abstract function required for the robot

Function			Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Manual
For task	Surface	Shape Detection			✓				✓		
		Color Detection	✓	✓	✓	✓			✓	✓	
		Relative Pos. Detection	✓	✓	✓	✓			✓	✓	
		Ball launch system							✓		
		Sound source Detection								✓	
	Underwater	Shpae Detection					✓				
		Color Detection					✓	✓			
		Relative Pos. Detection					✓	✓			
Navigation		GPS	✓	✓	✓	✓	✓	✓	✓	✓	
		Generating course	✓	✓	✓	✓	✓	✓	✓	✓	
Communication		Wi-fi	✓	✓	✓	✓	✓	✓	✓	✓	✓
Safety		Emergency stop(man.)	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Emergency stop(remote)	✓	✓	✓	✓	✓	✓	✓	✓	✓

Fig. 1: LRF:HOKUYO YVT-X002

Fig. 2: Stereo Cameras: WATEC WAT- 02U2D

50 [m] will be required to detect obstacles and generage environment maps, and the density of the sample points should be higher.But a LRF with such ability cost more than 10,000 USD, and it is impossible for us to purchase.So we decided to use LRF from our sponsor but it is not too strict, and then, we adopt visual odometry system to compensate for the function of LRF.
For visual odometry, we need stereo camera.The scale of the robots is larger than normal grand robots,so distance beteween two cameras is necessary to be as wide as possible.And camera have to be waterproofed.So we combine two waterproof cameras and use as a stereo(Fig.2).

2.2 Computer and Algorithm

The last competition, we used windows7 as the operating systems for our robots.However,it is difficult to control by Remote-Computing.The controlbility from remote place is very important in this tasks, so this time we considered to use terminal SSH.We adopted Linux OS.The number of programmer is not so sufficient in our team, we have to develop all systems for short time. So we tried to use MATLAB and ROS(Robot Operating System) with some packages.MATLAB is helpful for us to generate environment maps with GPGPU technology: CUDA library of Nvidia Corporation.The stable version of ROS was Indigo and it is for Ubuntu14.04.Then,we decided to use Ubuntu14.04 LTS, Matlab 2016a, and ROS Indigo for developing. Also we used python3 for some macro.

2.3 Hardware

In the last competition, the power of motors on our machines was too weak to control heavy WAM-V.So this time we would like to equip more powerful motor.In 2014, our systems are drived by 12V,but the more power if motor output, the more current is required.So this time, we develop power supply system with 24 volts and reduce current.And this way is helpful to expand the capacity of the battery without they connect pararell; because battery pararell connection should be avoid.

Also, we discussed which signal protocol should be used between computers and all other equipments.In the previous section, we decided all components required to equipe on the robots.So next,we draw the components connection diagram and argued the protocol.In the end, we adopted USB-serial, and Ethernet(UDP).Most of the sensors already had their own protocol, and we decided to use them without any modification.The length of transporting signals is not so long, so both USB-serial and Ethernet are certainsure we thought.Of course, CAN is also strong way of transmitting,but it is too difficult and too strong to develop in the limited time, so we abondoned this suggestion.Finally, we gathered all opinions and draw

connection diagram as Fig.5 The number of the members are very few, we have to operate robots with few people.So we considered that equipments can be put and constract easily for few operators. For example, all of the equipments can be connected with water proof connectors.

3 Hardware Design

3.1 Computing Units

In our strategy, we will use High-End GPU(Nvidia GTX1060) for map generating,so our server are planned with below components,

•**CPU** intel Core i7- 6700

•**RAM** Corsair DDR4 32GB

•**GPU** Nvidia GTX1060

•**SSD** Fixstars 3TB (SATA3-6Gbs)

We selected highly efficient parts because the power source is limited on the ship.Then we set target maximum watt as around 300 [W].The computing devices have to be waterproofed,but this device will also become considerable hot, so we have to consider how to flow the air in the box too.To solve this problem, we searched how the outside device around the world eliminate hot air to outside.Finally, we encounted waterproof louvers.So we equiped them on the box and make it to avoid from thermal runway.We call this equipments as "Blue BOX." Inside looks like Fig3.We designed it not to interrupt air flow,by adjust length of cables,put devices linearly.And,external devices can connect easily with waterproofed connectors,and turn on from remote wired switch.It helps operate machines by few operators.

3.2 Power Units and Emergency systems

This time,we choose 24V as basic volts of our systems, because it can helps to reduce currents going through wires and then we can use more thin cables and connectors, and it can also helps to the capacity of batteries can be doubled.But in order to use 24V to our machines, there is a problem.The more batteries we put on,the worse ship's mobility become, and the weight limit may be over.So we calculate how much currents will be flowed in the ship's power lines, and we choose proper battery.We calculated the avarage of it on motor will be around 46 [A](86 [lbs] max.46 [A] of each and the rate of oparation estimates 50 [%]).On the other hands, server

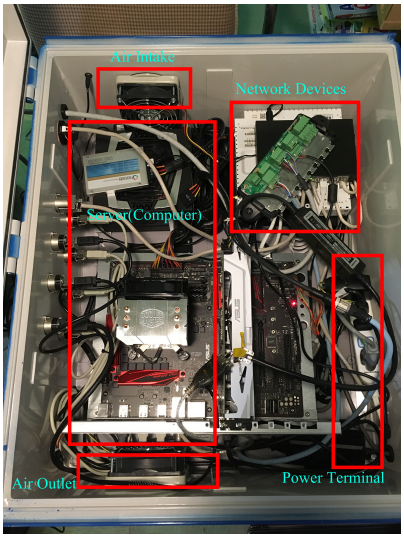


Fig. 3: The components in the Blue BOX

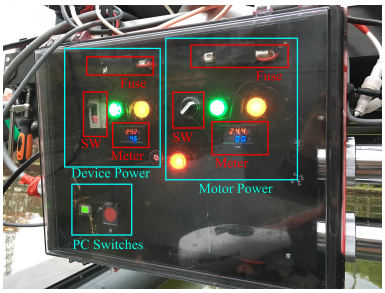


Fig. 4: Power Distribution Box

and other all electronical devices were estimated to be consumed around 220 [W](Max 400 [W] from description papers of each parts) and this estimates also help us to decide the DC24V/AC100V inverter's spec.
Our systems are mainly operated with 24 volts, but the load of motors will be change frequently, and it might adversely affect to operation of electrnical devices.So we divided the power unit into the power for control systems and motor systems.Each systems have to be turned on/off when we are necessary, and in the dangarous situations, motors systems should be forced to cut off physically.So we made pwer distribution box(We called it "Red Box.", Fig.4) included next functions: Hand switches, Emergency switches, Fuses(to avoid to burn), Indicator lights, and Distributing many types of electricity to each device(included connectors).We show the diagram of the box at Fig.6.And additonally, the switches and HDD lights of the server is eqipped too.

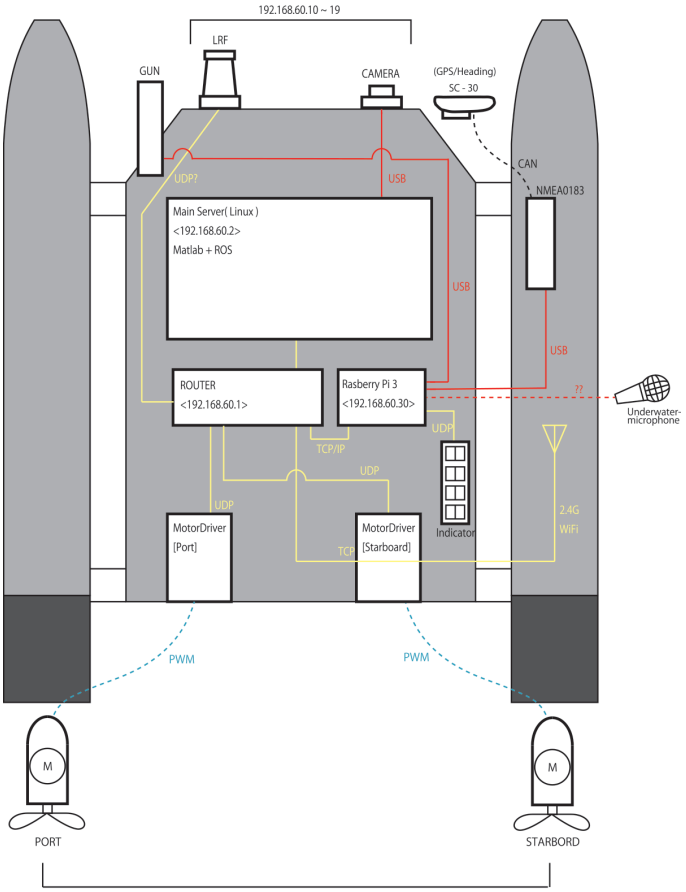


Fig. 5: Components Connection Diagram

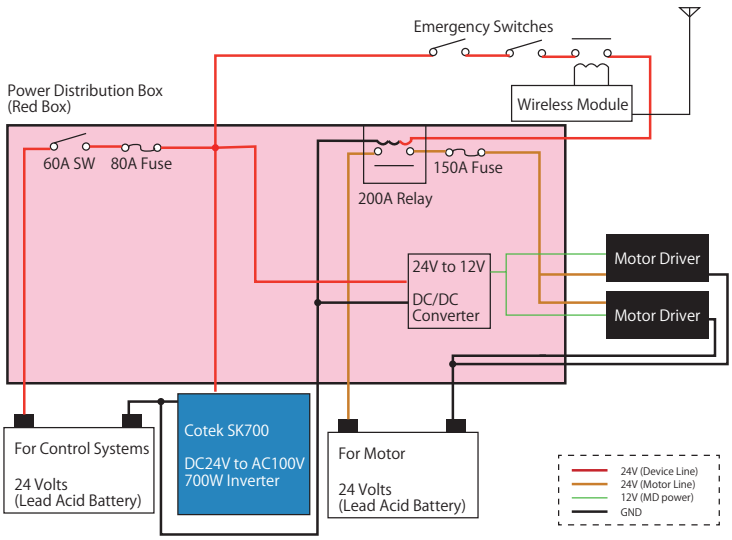


Fig. 6: Power Distribution Box Diagram

4 Algorithm Design

4.1 Dynamic model

4.1.1 WAM-V Dynamic model

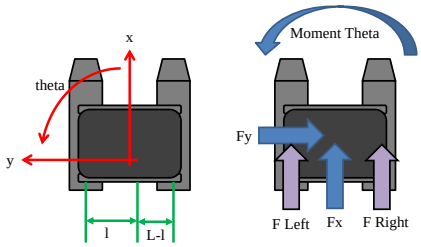


Fig. 7: Dynamic model of WAM-V

We make a 2D dynamic model of WAM-V[fig.7].The left image of fig.7 shows the coordinate which we defined and the right image of fig.7 shows a WAM-V's dynamic models. $Moment\,Theta(kgm^2)$ is the moment caused by external force. F_x[N] and F_y[N] are external forces. F_{Left}[N] and F_{Right}[N] are forces generated by the Left and Right motors. l[m] is the distance between the left motor and center of mass. L[m] is the distance between the right motor and left motor.
This dynamic model consists of below three mathematical equations.

$$m\frac{\mathrm{d}^2\mathrm{x}}{\mathrm{dt}^2} = F_x + F_{\mathrm{Left}} + F_{\mathrm{Right}} - K_x\frac{\mathrm{dx}}{\mathrm{dt}} \tag{4.1}$$

$$m\frac{\mathrm{d}^2\mathrm{y}}{\mathrm{dt}^2} = F_y - K_y\frac{\mathrm{dy}}{\mathrm{dt}} \tag{4.2}$$

$$I\frac{\mathrm{d}^2\theta}{\mathrm{dt}^2} = M_\Theta$$
$$+(L-l)F_{\mathrm{Right}} - lF_{\mathrm{Left}} - K_\theta\frac{\mathrm{d}}{\theta}\mathrm{dt} \tag{4.3}$$

m is WAM-V's mass[kg], and I is it's moment of inertia[kgm^2].K_x,K_y,K_θ is resistance coefficient of water.

4.1.2 Propeller and motor dynamic model

We make a very simple model of Propeller and motor dynamic model. We approximate relationship between propeller rotational speed and propeller power like this[eq.4.4].

$$F = \exp(ar) \tag{4.4}$$

r is rotational speed of propeller[Hz], a is coefficient and F is force[N].
We control motor rotational speed by PWM wave. So, the relationship between propeller rotational speed and PWM Duty Ratio is shown as the mathematical equation below.

$$r = r_0 d \tag{4.5}$$

r_0 is maximam rotational speed[Hz], and d is duty ratio of motor[%]. Considering equation:4.4, equation:4.5, the propeller and motor dynamic model is approximated in this mathematical equation.

$$F = \exp(\mathrm{Pd}) \tag{4.6}$$

P is a coefficient.

4.2 Controller

4.2.1 2DoF Controller

2DoF controller is known as a ISA-PID controller. It is combination of Feedfoward controller and Feedback Controller(PID Controller).

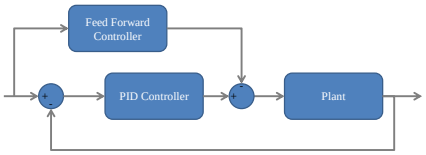


Fig. 8: 2DoF Controller

We use 2DoF PID Controller in order to converge controlled variable(longitudinal velocity, angular velocity) quickly. We also use 2DoF PID Controller to improve robustness of our systems. Feed foward controller provides input values calculated from the dynamic model of our vehicle. It is very difficult to measure m and I, so we make online estimation method which calculates parameters of dynamic model.

Online parameter estimation

From equation:1∼3, errors of 2DoF controllers are shown as these equations.

$$e_x = \dot{x}(t) - refV_{\mathrm{L}}$$
$$= \frac{1}{m}\int_0^t (F_x + F_{\mathrm{Left}} + F_{\mathrm{Right}})\mathrm{dt}$$
$$-refV_{\mathrm{L}} - K_x \tag{4.7}$$

$$e_y = \dot{y}(t) = \frac{1}{m}\int_0^t F_y\mathrm{dt} - \mathrm{K_y} \tag{4.8}$$

$$
\begin{aligned}
&e_\theta = \dot{\theta}(t) - refV_{\mathrm{a}} \\
&= \frac{1}{I}\int_0^t (M_\Theta + (L-l)F_{\mathrm{Right}} - lF_{\mathrm{Left}})dt \\
&-refV_a - K_\theta
\end{aligned} \tag{4.9}
$$

$refV_{\mathrm{L}}$ is reference longitudinal velocity, and $refV_{\mathrm{a}}$ is reference angular velocity.
In a Discrete system, equation:4.7∼4.9 is transformed to these equations(eq:4.10∼4.12).

$$
\begin{aligned}
&e_x = \dot{x}(t) - refV_{\mathrm{L}} \\
&= \frac{1}{m}\sum_0^t (F_x + F_{\mathrm{Left}} + F_{\mathrm{Right}})\mathrm{d}t \\
&-refV_{\mathrm{L}} - K_x
\end{aligned} \tag{4.10}
$$

$$
e_y = \dot{y}(t) = \frac{1}{m}\sum_0^t F_y \mathrm{d}t - K_y \tag{4.11}
$$

$$
\begin{aligned}
&e_\theta = \dot{\theta}(t) - refV_{\mathrm{a}} \\
&= \frac{1}{I}\sum_0^t (M_\Theta + (L-l)F_{\mathrm{Right}} - lF_{\mathrm{Left}})\mathrm{d}t \\
&-refV_a - K_\theta
\end{aligned} \tag{4.12}
$$

External forces cause errors of controlled values, so minimizing influence by external forces on each steps is important. In order to do so, we calculate virtual mass and moment of inertia. WAM-V[eq:4.13∼4.16]. dt is time step of this discrete system[sec].

$$
\begin{aligned}
refV_{\mathrm{L}}\hat{m} &= (refV_{\mathrm{L}} + K_x)m - F_x\mathrm{d}t \\
&= (F_{\mathrm{Left}} + F_{\mathrm{Right}})\mathrm{d}t
\end{aligned} \tag{4.13}
$$

$$
\begin{aligned}
&refV_{\mathrm{a}}\hat{I} \\
&= (refV_{\mathrm{a}} + K_\theta)I - (M_\Theta)\mathrm{d}t \\
&= ((L-l)F_{\mathrm{Right}} - lF_{\mathrm{Left}})\mathrm{d}t
\end{aligned} \tag{4.14}
$$

$$
\hat{m} = \left(1 + \frac{K_x}{refV_{\mathrm{L}}}\right)m - \frac{F_x\mathrm{d}t}{refV_{\mathrm{L}}} \tag{4.15}
$$

$$
\hat{I} = \left(1 + \frac{K_\theta}{refV_{\mathrm{a}}}\right)I - \frac{(M_\Theta)\mathrm{d}t}{refV_{\mathrm{a}}} \tag{4.16}
$$

Algorithm 1 Update m

1: **while** WAM-V is working. **do**
2: Move WAM-V and measure errors.
3: Calculate external forces from errors.
$e_x = \dot{x}(t) - refV_{\mathrm{L}} = \frac{1}{m}\sum_0^t (F_x + F_{\mathrm{Left}} + F_{\mathrm{Right}})dt - refV_L - K_x$
4: Update K_x
$K_x = K_x + \alpha(F_x(t-1) - F_x(t))$($\alpha$ is coefficient)
5: Update m
$m = m + \beta(F_x(t-1) - F_x(t))$($\beta$ is coefficient)
6: Update $\hat{m}$
$\hat{m} = \left(1 + \frac{K_x}{refV_{\mathrm{L}}}\right)m$
7: **end while**

Parameter update rules

We cannnot get exact parameters of dynamic models, so we have to update parameters sequentially. If the estimation goes well, the second terms of equation:[4.15, 4.16] becomes smaller. So we made the parameter update rules below.

Algorithm 2 Update I

1: **while** WAM-V is working. **do**
2: Move WAM-V and measure errors.
3: Calculate external forces from errors.
$e_\theta = \dot{\theta}(t) - refV_{\mathrm{a}} = \frac{1}{I}\int_0^t (M_\Theta + (L-l)F_{\mathrm{Right}} - lF_{\mathrm{Left}})\mathrm{d}t - refV_{\mathrm{a}} - K_\theta$
4: Update K_θ
$K_\theta = K_\theta + \alpha(F_x(t-1) - F_x(t))$
(α is coefficient)
5: Update I
$I = I + \beta(F_x(t-1) - F_x(t))$
(β is coefficient)
6: Update $\hat{I}$
$\hat{I} = \left(1 + \frac{K_\theta}{refV_{\mathrm{a}}}\right)I$
7: **end while**

4.2.2 Pure Pursuit Controller

Pure Pursuit Controller is one of the path following algorithm,and it generates the longitudinal velocity and angular velocity of our vehicle[1]. Pure pursuit controller is often used in autonomous car control algorithm[2].

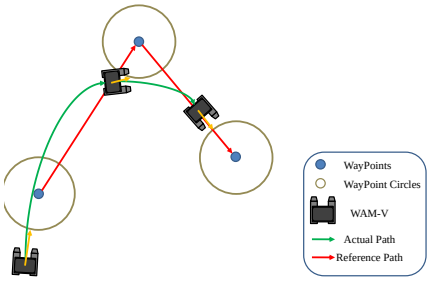


Fig. 9: Pure Pursuit Controller

Pure Pursuit Controller calculate look-ahead-distance from each waypoint circles(length of yellow arrow in fig[9]). Then, the controller calculate longitudinal velocity and angular velocity from look-ahead-distance vector(yellow arrow in fig[9]).

4.3 Navigation

Navigation system of our vehicle consists of these elements(fig:10).

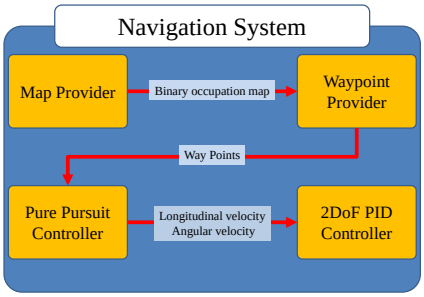


Fig. 10: Structure of navigation system

Map provider provide Binary occupation map to Waypoint provider. Waypoint provider provide Waypoints to Pure Pursuit Controller. Pure Pursuit Controller calculates longitudinal velocity and angular velocity to 2DoF PID Controller. 2DoF PID Controller calculates dynamic model of WAM-V and generates low-level control signals(PWM ratio for right and left motors).

4.4 Image Processing

4.4.1 Filter

We make filters in order to detect buoys in camera images. By using this filter, our vehicle can avoid misrecognition of buoys. This algorithm works before our vehicle runs the buoy detection and tracking algorithm.
This filter includes winner filter and OTSU method. Winner filter helps us to make adaptive filters for noises(water surface glance, background image, sunshine, etc...), and OTSU method can decide threshold of Saturation.

Adaptive filtering for extract buoy areas in images.

Contents

- Load Image and Pre processing
- Use winner filter
- Extraction

Load Image and Pre processing

```
% Load image
img = imread('../sea.jpg');

%  Resize Image
img = imresize(img,[480,640]);
figure;
imshow(img);
xlabel('Images before extraction','FontSize',16);
```

Images before extraction

Use winner filter

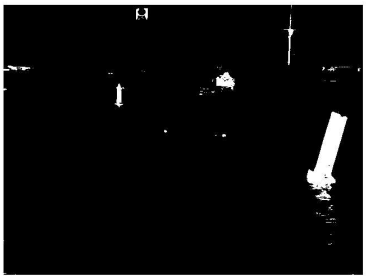
```
% winner filter Window Size
WindowSize = [20,20];
img(:,:,1) = wiener2(img(:,:,1),WindowSize);
img(:,:,2) = wiener2(img(:,:,2),WindowSize);
img(:,:,3) = wiener2(img(:,:,3),WindowSize);
figure;
imshow(img);
xlabel('Images processed py winner filter','FontSize',16);
```

Images processed py winner filter

Extraction

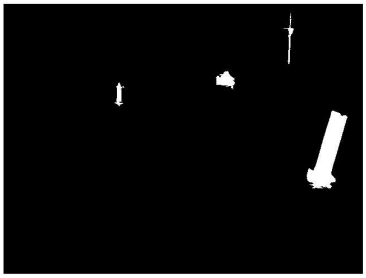
```
% rgb to hsv
img = rgb2hsv(img);

% Use Otsu methoc in Saturation
th = graythresh(img(:,:,2));
mask = im2bw(img(:,:,2),th);
figure;
imshow(mask);
xlabel('Mask Image','FontSize',16);
```



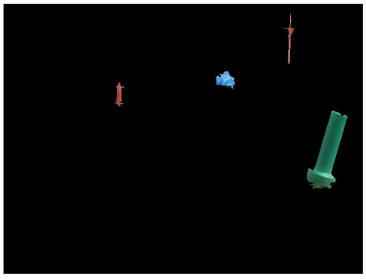
Mask Image

```
%  Remove small areas
mask = GetArea(mask);
figure;
imshow(mask);
xlabel('Images after removed small areas.','FontSize',16);
```



Images after removed small areas.

```
%  Mask images.
mask = double(mask);
img(:,:,1) = img(:,:,1) .* mask;
img(:,:,2) = img(:,:,2) .* mask;
img(:,:,3) = img(:,:,3) .* mask;
% hsv to rgb
img = hsv2rgb(img);
figure;
imshow(img);
xlabel('Output Image','FontSize',16);
```



Output Image

Published with MATLAB® R2016b

4.4.2 Buoy detection

Our bouy detection system refer to mathworks demo.[3] The buoy detection is based on RGB values and shapes in each areas that are extracted by filters in previous section. First, we calculate avarage of RGB values in each extracted areas. Second, if R values are in our set range, we decide this areas as candidate areas of red buoys, and if G values are in our set range, we detect this areas as candidate areas of green buoys. Then, we try to find areas that have very high R, G, and B values. We decide that areas as candidate areas of white buoys. After that, draw rectangles of detected areas and calculate aspect ratio. If the aspect ratio is near to buoy's aspect ratio, we decide that areas as buoy detected area.

Algorithm 3 Buoy detection

while Wam-V is working. **do**
  Get Image
  Pass Filters
  Calculate avarage of RGB values in each extracted areas.
  if $threshold_high > R > threshold_low$ **then**
    Decide this areas as candidate areas of red buoys.
  end if
  if $threshold_high > G > threshold_low$ **then**
    Decide this areas as candidate areas of green buoys.
  end if
  if $threshold_high > G > threshold_low$ and $threshold_high > R > threshold_low$ and $threshold_high > G > threshold_low$ **then**
    Decide this areas as candidate areas of buoys.
  end if
  Calculate aspect ratio of rectangles of candidate areas.
  if Aspect ratio of rectangles of a candidate area $\approx$ Buoy's aspect ratio. **then**
    Find buoy!!
  end if
end while

4.4.3 Localization

We make rectangles of buoys in previous section(Section 4.4.2). And we have a stereo camera on our vehicle, so we can get parallax image of buoys. So, we can estimate the distance between our vehicles and buoys.

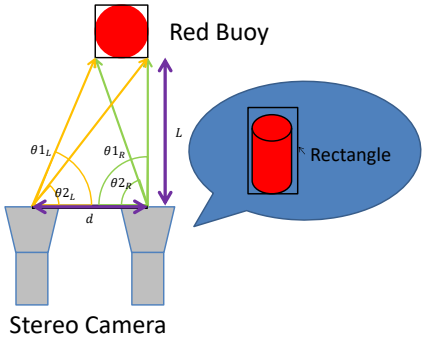


Fig. 11: Buoy localization

Fig.11 shows that L and another values have this relationship.

$$d = \frac{L}{sin(\theta 1_L)} + \frac{L}{sin(\theta 1_R)} = \frac{L}{sin(\theta 2_L)} + \frac{L}{sin(\theta 2_R)} \tag{4.17}$$

So, L is,

$$\begin{aligned} L &= \frac{d(sin(\theta 1_L) + sin(\theta 1_R))}{sin(\theta 1_L)sin(\theta 1_R)} \\ &= \frac{d(sin(\theta 2_L) + sin(\theta 2_R))}{sin(\theta 2_L)sin(\theta 2_R)} \end{aligned} \tag{4.18}$$

5 Experiment

We did some experiment on the water several times.However, there was no lift or crane besides the pond in the university, and there are few members in our team.So this time, we couldn't do sufficient experiment now.In the several experiments, we confirmed hardwares can be operated without any problems.So later, we will find any proper place and do experiments and confirm that our algorithms follow to our designed and theories.

References

1) MathWorks,"Pure Pursuit Controller", http://jp.mathworks.com/help/robotics/ug/pure-pursuit-controller.html

2) Shinpei KATO,Eijiro TAKEUCHI,Yoshio ISHIGURO,"AN OPEN APPROACH TO AUTONOMOUS VEHICLES", IEEE Micro 35 (6), 60-68, 2015

3) MathWorks,"MATLAB and Simulink Unmanned Vehicle Systems: Buoy Detection Using Simulink", https://jp.mathworks.com/videos/buoy-detection-using-simulink-89114.html