Fall 2014

Bio-Inspired Robotic Fish With Vision Based Target Tracking

Eric Anderson
Purdue University

Graduate School Form 30
(Revised 08/14)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Eric Anderson

Entitled
Bio-Inspired Robotic Fish With Vision Based Target Tracking

For the degree of Master of Science in Mechanical Engineering

Is approved by the final examining committee:

Xinyan Deng

Justin Seipel

Reuben Goforth

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification/Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Xinyan Deng

Approved by: Ganesh Subbarayan 12/03/2014

Head of the Department Graduate Program Date

BIO-INSPIRED ROBOTIC FISH WITH VISION BASED TARGET TRACKING

A Thesis

Submitted to the Faculty

of

Purdue University

by

Eric J. Anderson

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Mechanical Engineering

December 2014

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

This thesis would not have been possible without the help and support of a number of people, to whom I am greatly indebted.

Hilary, for her never ending encouragement and patience.

My parents, for raising me to love science and the pursuit of knowledge as well as instilling in me a passion for working with my hands. They have provided invaluable advice and support throughout my educational career, especially my graduate studies.

Finally, my adviser Dr. Deng and all my past and present lab mates. They have provided guidance, training, and invaluable experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Anderson, Eric J. M.S.M.E., Purdue University, December 2014. Bio-Inspired Robotic Fish With Vision Based Target Tracking. Major Professor: Xinyan Deng, School of Mechanical Engineering.

The lionfish is an invasive species that out-competes and overcrowds native fish species along the eastern seaboard of the United States and down into the Caribbean. Lionfish populations are growing rapidly. Current methods of monitoring lionfish populations are costly and time intensive. A bio-inspired robotic fish was built to use as an autonomous lionfish tracking platform. Lionfish are tracked visually using an onboard processor. Five different computer vision methods for identification and tracking are proposed and discussed. These include: background subtraction, color tracking, mixture of Gaussian background subtraction, speeded up robust feature (SURF), and CamShift based tracking. Each of these methods were compared and their accuracy analyzed. CamShift based tracking is determined to be the most accurate for this application. Preliminary experiments for system identification and control design are discussed.

1. INTRODUCTION

This project was motivated by a problem that has recently gained considerable attention in the marine biology community. The lionfish (*Pterois*)is a venomous reef fish native to the tropical Western Pacific. However, due most likely to an accident release from an aquarium, they are now found along the eastern seaboard of the United States and into the Caribbean [1]. In the ten years since first being observed in North American waters, the population has exploded, with surveys finding lionfish to be the second most abundant species in waters from Florida to North Carolina [2], as well as greater than 300 lionfish per hectare in the Bahamas region [3]. This is a cause of concern due to the high impact that lionfish populations have on native fish colonies [4]. Because to their highly predatorial nature and lack of predators they adversely effect native ecosystems through competition and overcrowding [2]. They are considered to be a highly invasive species, often compared to the zebra mussel invasion of the Great Lakes.

Collaborating with Dr. Reuben Goforth in the Purdue University Department of Forestry and Natural Resources our goal was to design and construct an autonomous underwater vehicle (AUV) capable of tracking lionfish and gathering population data for use by biological researchers. Current methods of population monitoring and tracking are limited to SCUBA divers and remote operated vehicle (ROV) transect surveys. These methods are costly and time consuming and not necessarily accurate [6]. Our goal is to give marine biologists another tool to collect this population data, and help fight invasive species. To do this our system had to meet specific requirements. To have the maximum utility for researchers, fully autonomous operation was necessary. In order to not disturb the territorial lionfish it needed to be non-threatening and have low noise during operation. Most importantly, it is requires the ability to accurately identify and track lionfish.

Figure 1.1: A Red Lionfish [5].

There are two facets to this project. First is the design and construction of the robotic platform itself. The shape of the body, actuation of the fins, and electronics package must all be carefully designed to achieve the goal of autonomous fish tracking. The second facet is the method of detecting and identifying lionfish. Because the robot will operating in an unstructured reef environment, this detection method must be highly robust. A biologically inspired AUV with vision based navigation was chosen to meet these requirements.

2. BACKGROUND

Extraordinary strides have been made in the field of AUV's and bio-mimetic underwater robotics. Most of these advancements are in the fields of propulsion, control, and sensing. In the last decade or so, many of these have come about as a result of taking engineering inspiration from nature [7]. Therefore, in order to understand the engineered systems, it is helpful to have a basic knowledge of the biological systems they are based on. The following sections give a brief overview of underwater locomotion, the role of vision for fish, and several key examples of the state of the art in the field of autonomous underwater vehicles, most especially those with vision systems, and biologically inspired underwater robots.

2.1 Underwater Locomotion

Fish swimming is a very complex problem. The interactions between water, fish body, and fins are difficult to model and make the control of robotic fish a challenging problem [8]. As engineers there is much we can learn from fish about underwater propulsion and maneuvering. Biological systems are, in general, more efficient and more maneuverable than the current man made state of the art [7].The first step towards understanding and emulating biological fish is to examine how they propel themselves.

In general, underwater fin-based propulsion is divided into two categories [9].

1. Body and/or caudal fin motion (BCF)
2. Median or paired fins propulsion (MPF)

In a general sense, BCF motion is when a fish uses its caudal fin or some rear portion of its body as the main source of propulsive force [10,11]. An example of this

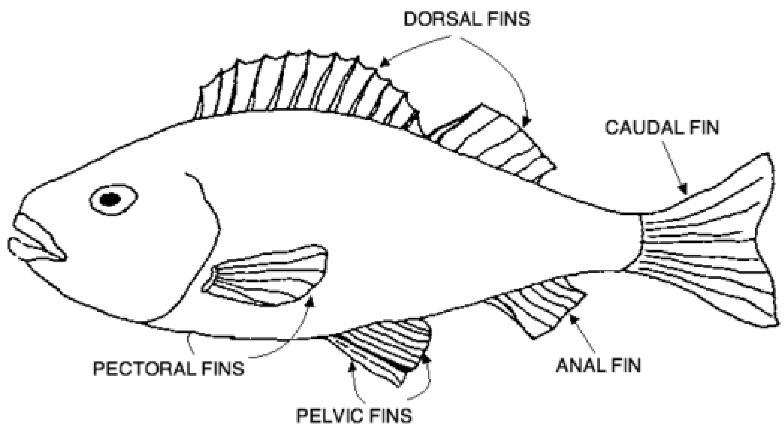

Figure 2.1: General Fish Morphology [8].

is the bluefin tuna or great white shark. MPF motion is based on the motion of the body fins. Most often the pectoral pins in combination with the the anal and dorsal fins [10].

Both of these types of motion then fall on a spectrum from undulatory to oscillatory motion, and can be further classified based on the amount of body that is used for propulsion in BCF and the pairs of fins used in MPF. Fig. 2.2 Both types of motion have advantages and disadvantages. Many fish can change where on the undulatory to oscillatory spectrum they fall and even switch between BCF and MPF to some extent [9]. For example, thuniform locomotion is characterized by using only the tail fin and flexible body for propulsion. It is very efficient for high-speed, long-range cruising, such as ocean tuna do. However, thunniform motion is not very effective in environments that require high maneuverability, quick starting and stopping, or small turning radii [10, 11].

Ostraciiform motion is fully oscillatory BCF propulsion. This type of motion is usually used by fish with stiff, inflexible carapaces and is combined with MPF propulsion using the pectoral fins to maneuver and hunt for food in a cluttered reef environment. The boxfish (Ostraciidae) is one example of a species that uses this technique [10]. The boxfish is commonly found in the coral reefs of the Indo-Pacific region. Boxfish use MPF motion for most speed ranges. At low speed ranges a com-

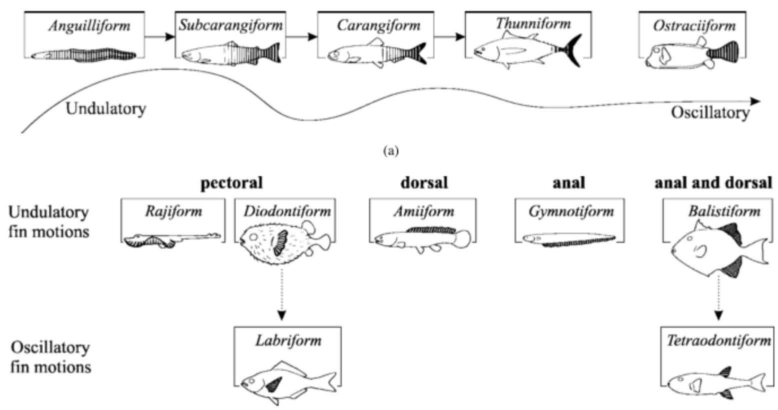

Figure 2.2: Classification of BCF (top) and MPF (bottom) swimming modes [9].

bination of pectoral and anal fins is used, at higher speeds, bursts of caudal fins beats are used [12]. Speeds over 6 BL/s can be achieved using this technique. Additionally, it has been shown that the rigid ventral keels of the boxfish carapace control vortex develop in several species. This effectively acts as a passive self-stabilization system for both pitch and yaw [8, 13, 14]. This is an important evolutionary advantage. Boxfish typically reside in turbulent reef environments; this vortex shedding mechanism helps them to efficiently maintain their position and orientation while foraging for food in the reef without expending additional energy. Additionally, because the mechanism is passive and automatic, it requires no neural processing [14]. This is important in the often unpredictably turbulent reef waters where a powered control system would require rapid processing.

However, there are disadvantages. The hard, boxy body produces high drag. Maneuvering is also more costly. Any course change, either from an environmental disturbance or by design, is counteracted [14]. However, there is evidence that pectoral fin motion is used as a powered stability mechanism [15]. Boxfish have the ability to cancel out or augment the body induced vortices. Therefore, the passive stability mechanism could be overruled when greater maneuverability is required [16]. This requires a greater energy expenditure than if vortices did not need to be countered.

This passive stability is a very attractive feature for an AUV. It would enable lower powered station-keeping and help to keep the platform stable while using a vision system. For these reasons, a box-fish like outer shell was chosen for the design of this robotic fish.

2.2 Underwater Sensing and Vision

Fish are able to swim in large schools and execute fast, complex schooling maneuvers without collisions. These maneuvers require fish to keep individual distances as well as maintain relative velocities by responding rapidly to their neighbors changing direction and speed. Vision and lateral line sensing form the sensory basis for this schooling activity [17].

It has been found that fish that had their lateral lines impaired could compensate using vision, and blinded fish were also able to maintain matching velocities with neighboring fish [17]. Maintaining a constant distance from neighbors is also a function of both vision and lateral line sensing. When fish are blinded, distance decreases, and if lateral line function is removed, distance increases [17]. It is thought that visual stimuli is attractive and lateral line stimuli is repulsive, the balance of these two is what determines the distance a fish keeps from its neighbors in a schooling environment [18].

The role of vision changes based on the habitat a species of fish resides in. Dark adapted fishes rely more heavily on lateral line, olfaction, and taste, while light adapted fishes rely more heavily on vision. In general however, fish hunt visually and rely most heavily on their eyes for finding prey [19]. Feeding efficiency is dependent upon light intensity. This influences the performance of the eye and depends on time of day, time of year, depth, line of sight, and optical clarity of the water. Most fish are thought to have color vision, and some species are sensitive to polarized light [20]. Many species of fish can also see lower into the infrared or higher in to the ultraviolet ranges than humans [21]. Carp, for example, have cone cells in theirs eyes that are specifically sensitive to ultraviolet and polarized light [22]. In short, vision is a very important sense for underwater life. The high reliance that fish have on vision, especially while hunting, indicates that it has a high potential for being a robust navigation and sensing tool for underwater robotic platforms. When developing underwater bio-inspired robots, vision can be a very useful sensory input. This lionfish tracking application is a particularly good application for the implementation of a vision guidance system. This is because lionfish general live in shallow reef areas where there is clear water and good light penetration. Lionfish also have a distinctive coloring and shape that make them stand out visually.

2.3 Previous Work

Much work has already been done in the field of underwater robotic vehicles. The recent advancement of onboard and embedded computer power has enabled an increase in capabilities, including more advanced control methods and navigation using computer vision. The trend in the last decade towards bio-inspired design has also brought about considerable advancement in the field [7]. Novel methods of propulsion and actuation, hull design, and sensing have all come about from trying to integrate lessons learned from biological systems into AUV design. Some key examples of recent work that our design builds are on outlined in the following sections.

2.3.1 Starbug AUV

Starbug is an autonomous underwater vehicle (AUV) designed by a team at the Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia's national science agency. Its purpose is to monitor and collect data on the Great Barrier Reef [23]. The reef covers over 349,000 km^2, so it is not practical for human divers and monitoring stations to adequately monitor the entire reef. Therefore, a vehicle was designed that could perform video transect surveys and monitor water quality [24].

Figure 2.3: The Starbug AUV [23].

Starbug has a mass of 26kg, a maximum forward speed of 1.5 m/s ,is fully actuated with six propeller thrusters giving forward, vertical, and lateral translation as well as roll, pitch and yaw, and has a sensor suite that includes two sets of stereo cameras, an inertial measurement unit, magnetic compass, pressure sensor, and GPS [24]. One set of stereo cameras points forward and one point downward. The downward cameras are used to estimate the altitude above the seafloor and velocity and the forward for obstacle avoidance [24, 25]. Starbug was one of the first vehicles to use autonomous vision navigation in the field.

The Starbug platform has been utilized for autonomous population monitoring of the crown-of-thorns starfish. The crown-of-thorns starfish (COTS) is an animal that feeds on the coral of the great barrier reef, killing it in the process. Large populations of COTS can quickly cause large amounts of damage to a reef. For this reason, it is desirable to monitor, track, and control COTS outbreaks [6]. However, this is a time intensive and error prone process when conducted by humans. Clement et.al. proposes a combination of robotic transect survey and automatic image recognition of COTS.

In order to segment the COTS from the background environment, a texture recognition algorithm was employed. This method was suitable because, while their color blends in very well with the environment, COTS have a distinguishable and recognizable thorny texture. This algorithm was able to correctly detect the presence of COTS in 65% of images, and able to find 77% of the total instances of COTS in an image where a starfish was detected [6].

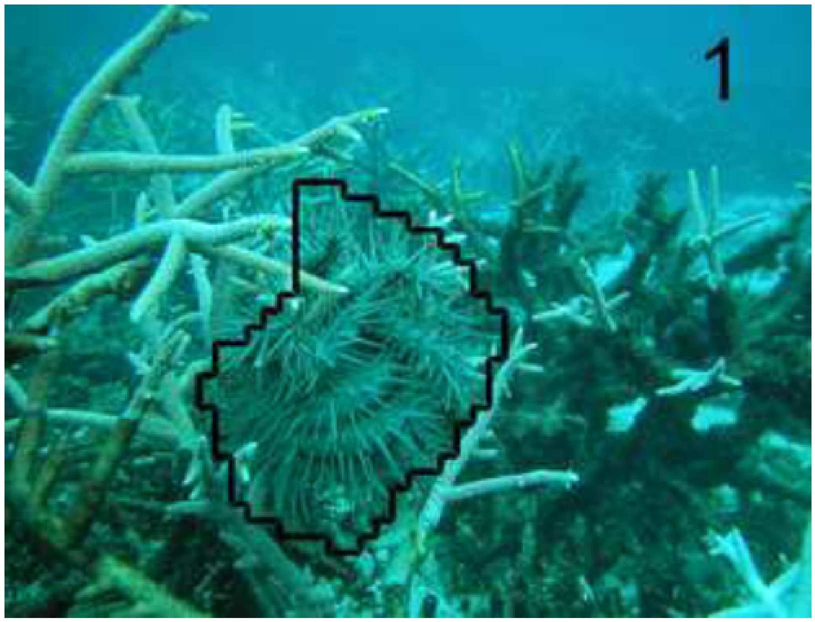

Figure 2.4: A correctly detected COTS [6].

This application is similar to our goal of lionfish tracking.Both applications want to achieve autonomous population monitoring by using an AUV equipped with a vision system. However, there are some key difference. Mainly, Starbug recorded video transect data that was later processed for COTS identification. This was not done onboard and was not used for navigation purposes. Secondly, Starbug is a traditionally designed AUV that, while effective, does not meet our requirements for a minimally invasive solution as well as a biologically inspired system would.

2.3.2 MARCO

Developed at the University of Delaware, the Micro Autonomous Robotic Ostraciiform (MARCO), is modeled on a boxfish. The body shape was designed to emulate the passive stability vortex shedding properties of real boxfish. In nature, the boxfish uses five fins to maneuver and propel itself. MARCO is propelled by two pectoral fins, each with two degrees of freedom, and a single degree of freedom caudal fin [26]. These fins are actuated by three servo motors held by parallel plates. A coaxial wrist mechanism is used to control the pectoral fins. The caudal fin mechanism consists of a gear stage between the tail shaft and the motor shaft [27].

Size and weight			
Length	Width	Height	Weight
15cm	9.2cm	8.5cm	490gm
Fin transmission gear ratios			
G_t^{θ}	G_t^{ϕ}	G_w^{ϕ}	G^{ψ}
0.75	1.33	1	0.75

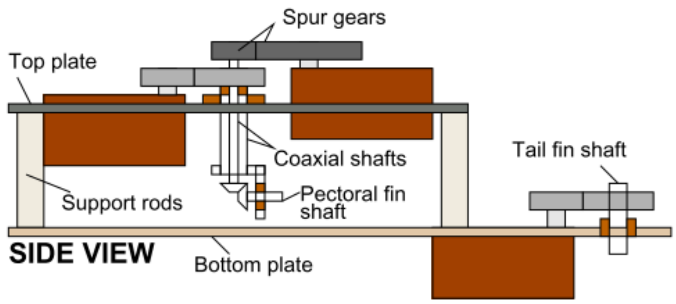

Figure 2.5: MARCO Parameters and chasis design [26].

The outer shell was 3D printed and then sealed using adhesive tape and latex film. This palm sized robot was able to obtain an average speed of .0411 m/s and an almost zero turning radius [28]. This was the first generation of robotic boxfish that led to the current design.

2.3.3 Vision Based Autonomous Robotic Fish

A boxfish-like robot capable of visual target tracking and following has been developed by Hu et.al. Similarly to the MARCO robot, it has a rigid body, two pectoral fins and a tail fin. However, this body is not designed with vortex shedding in mind and each fin has only one degree of freedom. Each fin is actuated by a servomotor that is connected to a microcontroller. The two pectoral fins have the approximate

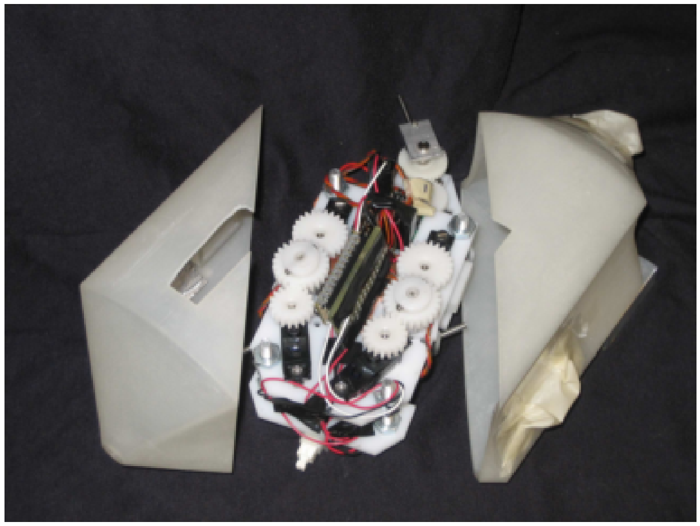

Figure 2.6: MARCO shell and gear train [26].

shape of NACA-0012 airfoil, and can be used as dive planes to change the depth of the fish [29]. The only sensor used is an image sensor mounted in the nose. The camera is connected to a wireless communication module through the onboard microcontroller unit. This allows the video to be transmitted to a base station for processing.

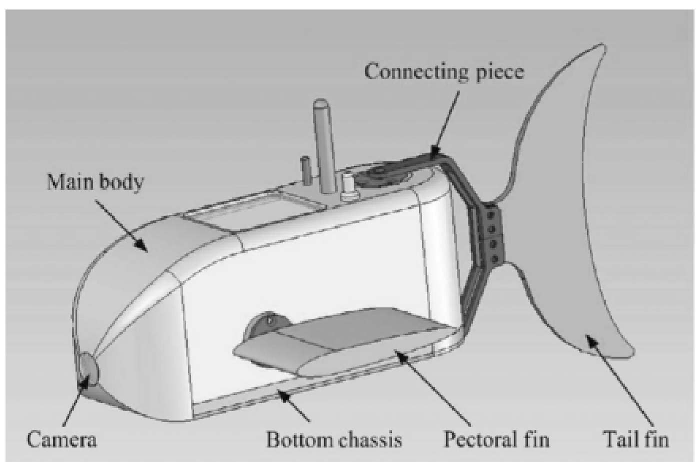

Figure 2.7: Configuration of the robotic fish [29].

The Camshift algorithm is used to track the a target. Distance to the target is determined by the height of the bounding rectangle generated by the algorithm. This visual information is then fed into a fuzzy logic controller which determines speed, gait, and orientation [29]. The tracking capability was demonstrated by tracking a second robotic fish, as seen in Fig. 2.8. While relatively straightforward, this method only works if the size of the object being tracked is previously known and used for calibration, otherwise a distance measurement cannot be achieved.

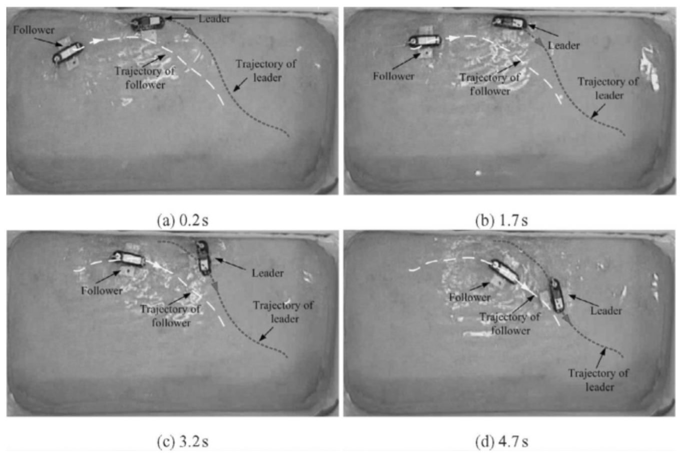

Figure 2.8: Target following demonstration [29].

This was an effective proof of concept that visual target tracking was able to be accomplished using an Ostraciiform based robot. However, lack of a fully developed sensor package and the use of pectoral fins with an inorganic airfoil-like cross section are drawbacks of this platform. Additionally, this work shows that Camshift works well in a laboratory testing environment, but does not demonstrate if it is feasible to use in a more natural environment or if tracking an organic fish as the target is possible. It also requires a base station for all image processing, which limits autonomy and range of the robot.

2.3.4 Fish Counting

The EcoGrid project in Taiwan collects real time ecological video data. Part of this system includes undersea cameras which record video 24 hours a day, 7 days a week. These cameras generate too much video to be analyzed by traditional means.

Spampinato et.al. propose an automated video processing system to detect and track fish the fish present in a video.

To detect the presence of a fish in the frame, a combination of moving average detection algorithm and the adaptive Gaussian Mixture Model were used. The Gaussian Mixture Model helps to eliminate the false positives that are a problem with the moving average detection method, and the moving average method props up the Mixture Model when it would fail due to slow moving objects [30].

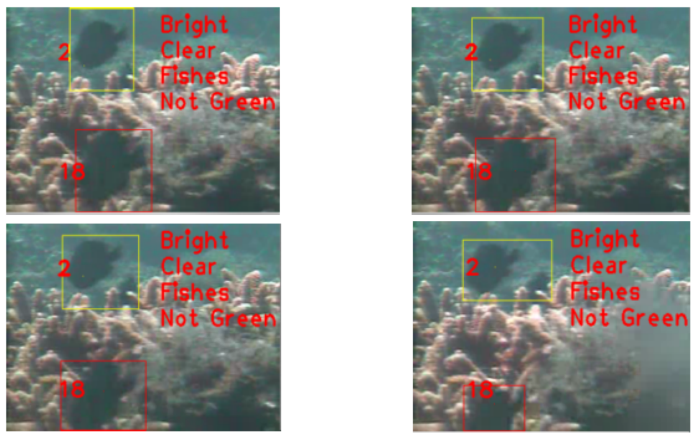

Figure 2.9: Fish being tracked in 4 consecutive frames [30].

Once a fish is detected, a tracking system is used estimate the total number of fish in the video. This tracking system consists of a combination of two algorithms, one based on blob shape matching and the other on color histogram matching [30]. The shape matching works by creating a feature vector for each blob of connected pixels and comparing these feature vectors in subsequent frames. The color histogram matching is completed using the CamShift algorithm, which is explained in detail later in this paper. This system achieves an average success rate of over 85%, which is quite good [30].

This method is of some interest to our current application. It seems to accomplish very similar goals of fish tracking. However, it uses stationary cameras and video processing is done off site after videos have been captured. Still, it demonstrates that accurate identification and tracking of fish in an unstructured environment is possible.

3. DESIGN AND CONSTRUCTION OF ROBOTIC PLATFORM

This chapter summarizes the design and construction of the robotic fish platform. This was a group endeavor that took place over the course of several years and included numerous iterations and prototypes. Continuous improvements are, in fact, on going. Contributing students included, but are not limited to: myself, Jian Zhang, Giovanni Barbera, Fan Fei,and Zhan Tu.

3.1 Design Criteria

As with the design of any product or device, the most important aspect to consider is that it is able to complete the task that the end user needs it to. In this case, the robot must be able to identify, track, and follow a lionfish in a reef environment. All of this should be done fully autonomously.

When stated simply, this seems like a straight forward task. However, there are many facets of the design that must be considered. The key systems include propulsion and actuation, sensing, power, and computation. Each of these systems must work in concert with the others. The necessity that all electrical components must be waterproof adds an additional layer of complexity. The author's main contributions were to the electronics systems, so that is what this paper will focus on.

3.2 Electronics

There are three main functional areas to the electronic systems of the robot.

1. Sensors

2. Processors

3. Motor Control and Actuators

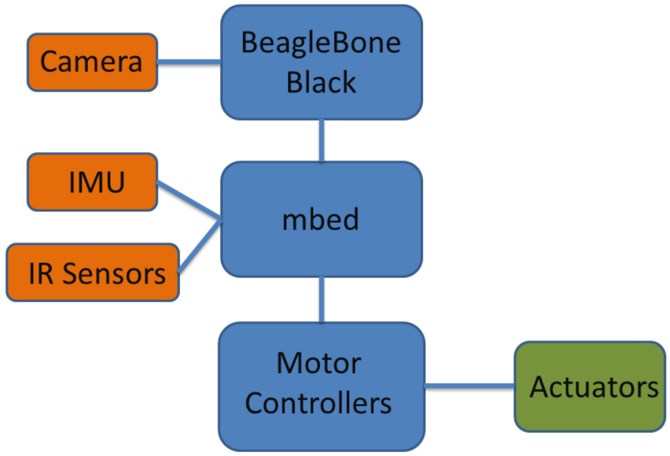

Figure 3.1: Electronics system block diagram.

At the core of the electronics system are the two processing units. These are the mbed and Beaglebone Black. The mbed is an ARM based microcontroller. The mbed package consists of the microcontroller hardware as well as an online compiler, function library, and database of user developed software. It has thirty I/O pins that can be used as serial, I2C, PWM, analog in/out, and more.

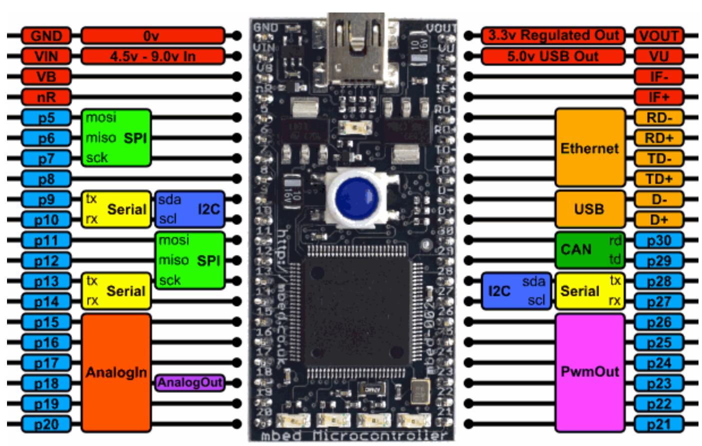

Figure 3.2: mbed pin diagram.

The Beaglebone Black (BBB) is a relatively new microcontroller released in mid-2013. It is a very capable board, with 512 MB of RAM, and processor speeds up to 1 GHz. It is capable of running a full version of linux. For our application, the Angstrom distribution is used. This distribution is advantageous because the OpenCV C++ vision library comes pre-installed.

The BBB has has two pin headers, each with 46 GPIO pins. The board also has an hdmi and usb port. These can be used to interface with a monitor and keyboard to program the board. A video camera can also be plugged in to the usb port, which is how it was used for our research. The board can be accessed through another computer using a terminal emulation program, such as PuTTY. This feature was heavily utilized during my research.

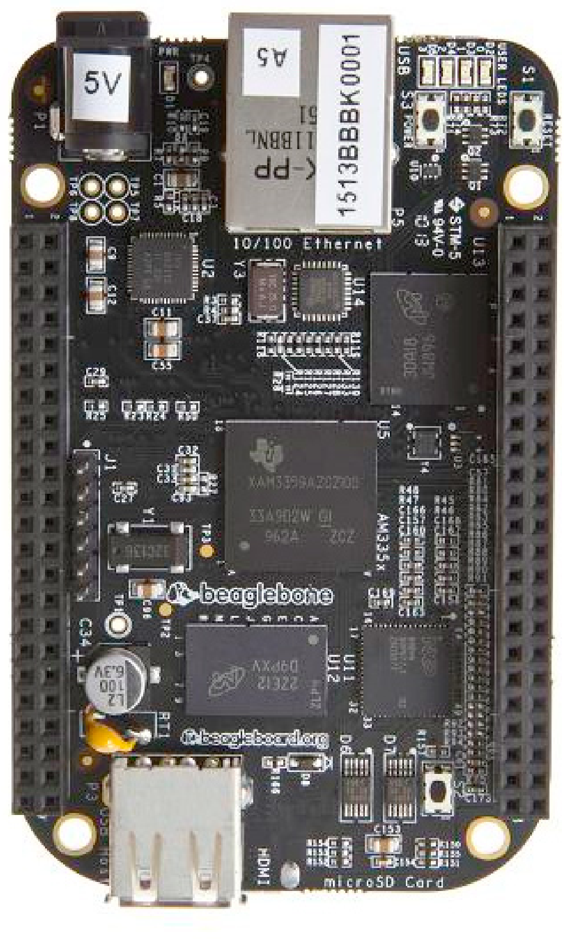

Figure 3.3: Beaglebone Black.

The BBB is used solely for the visual tracking portion of the guidance system. It has the requisite processing power that the mbed lacks for these types of calculations. However, it is not powerful enough to have the images display as it processes them. When attempted, this created significant lag in the processing and decreased the accuracy of the object detection and tracking program. This is the reason that the terminal emulation was used, it allows programming of the board without the need for a display.

As can be seen in Fig. 3.1, the BBB interfaces with the camera and mbed controller. The BBB processing the incoming video data to detect and identify lionfish. It then transmits the coordinates of the lionfish in the video frame to the mbed. The mbed interfaces with the other sensors and motor control boards. The other sensors in the system consist of an inertial measurement unit (IMU), and two infrared distance sensors. The inertial measurement unit includes a gyroscope, accelerometer, and magnetic compass. All controls operations are performed on the mbed. The control signal is then sent to the motor controller boards.

All electronics were integrated by using a custom designed printed circuit board. The PCB connected the mbed, BBB, motor controller boards, IR range finders, power supplies, and IMU. It also has pins available for additional sensors, such as a hydrophone and pressure sessor, a 12 and 5 volt power circuit, connections for servo motors, and an x-bee wireless communication module. This allows for future upgrades to the fish's electronics package.

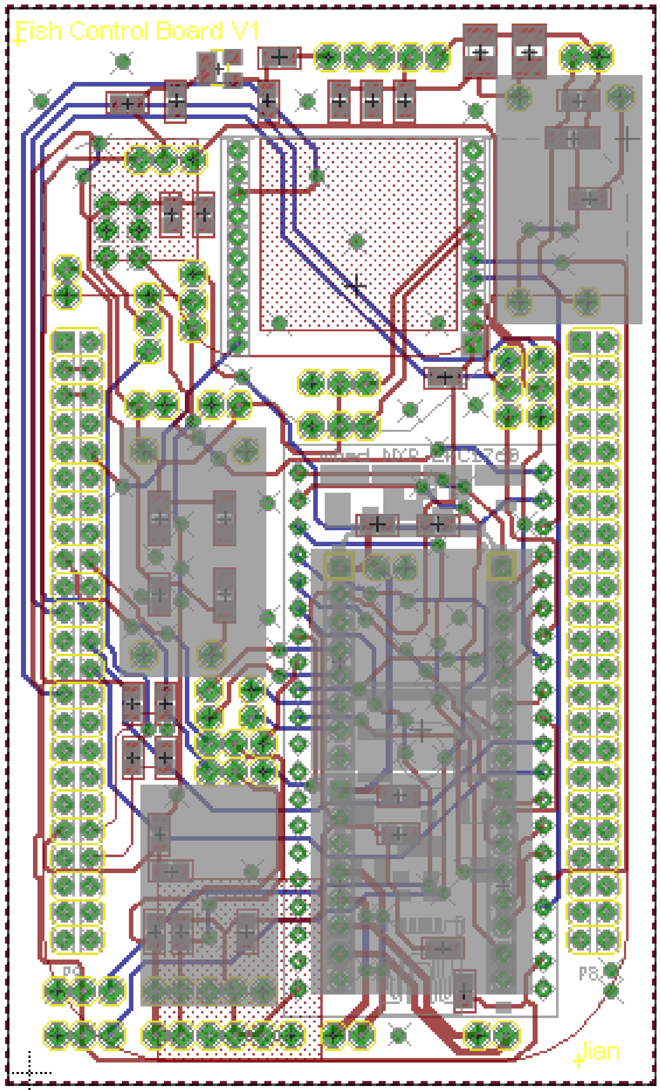

Figure 3.4: Printed Circuit Board design.

The system is powered by either 1 or 2 lithium polymer battery packs. Each pack provides 3.7V and 1200 mA. These are charged through a charging port or wirelessly using a charging coil and docking station. Each cell lasts approximately 45 minutes when all systems are operational.

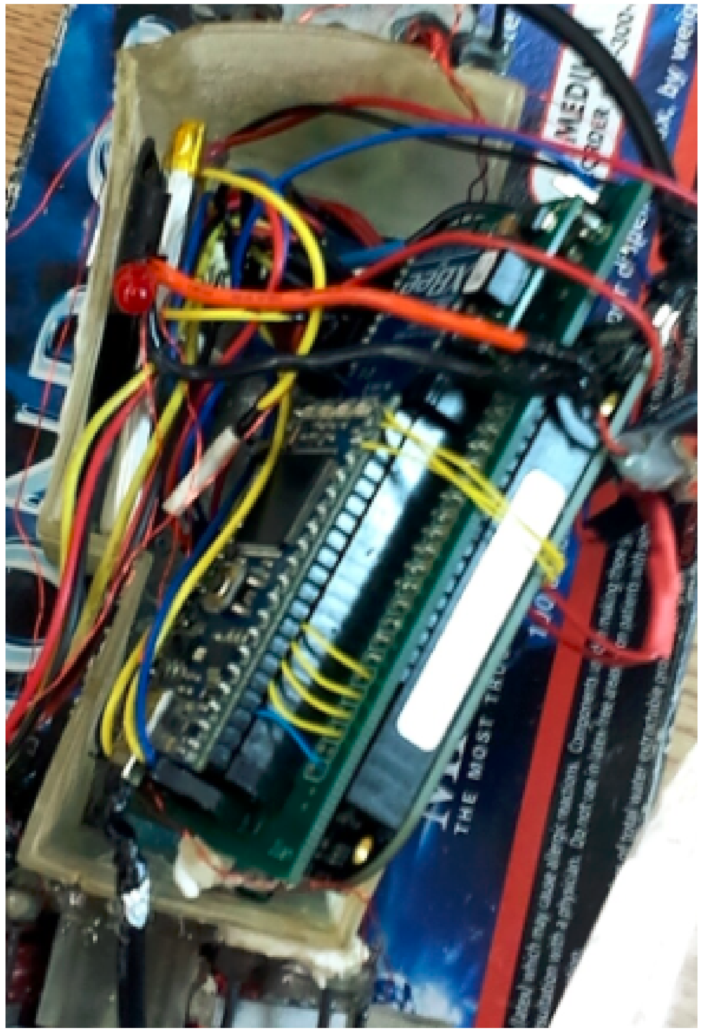

Figure 3.5: Electronic components being packed into the water proof shell.

3.2.1 Shell

The electronic components are housed in a waterproof enclosure. This enclosure is then covered in an external shell that emulates the shape of a boxfish. The internal water proof enclosure consists of a lid and box body. Both parts are 3D printed using rapid prototyping stereo-lithography technology. Ports for cables are left open. After the electronics package is installed in the box, cables for programming the mbed and beaglebone are threaded through the ports along with the leads for the actuators, IR sensors, and camera. These ports were then sealed 100% silicone caulking covered with a layer of epoxy. The silicone provides a degree of flexibility for the cable to move around slightly, and the epoxy layer provides a rigid outer shell that helps guarantee a good seal. A layer of epoxy was put around the edge of the lid where it met the box body. This made a water tight seal. However, if the cables were moved too much, the epoxy layer over the caulking could crack, letting water into the enclosure.

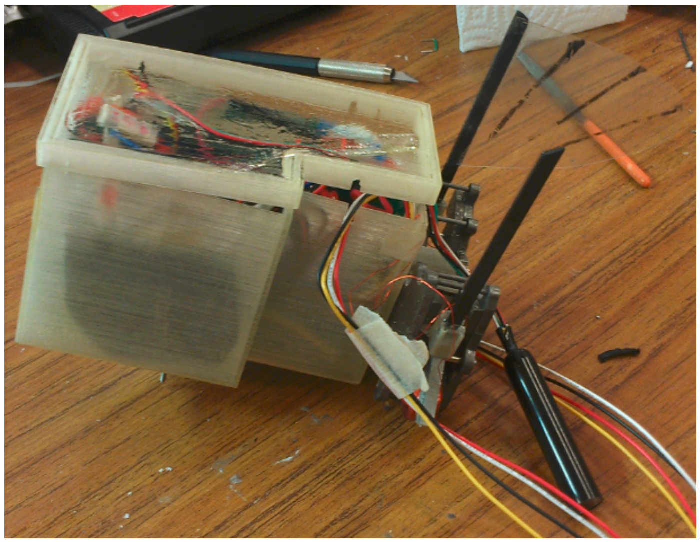

Figure 3.6: Electronic components in water proof shell.

Closed cell foam and weights were used to balance the robot to a neutral buoyancy in such a way that the fish floats upright. A magnetic leaf switch was attached just inside the lid, this allows the robot to be powered on and off by placing a magnet on the outside of the enclosure, eliminating the need for a switch that extends through the enclosure wall. This helps ensure a water tight seal is maintained. The custom designed electromagnetic actuators are also mounted on the outside of the electronics enclosure.

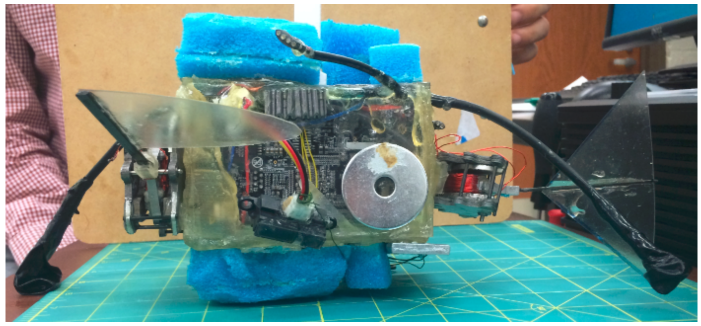

Figure 3.7: Electronic components in water proof shell with closed cell foam and actuators.

4. COMPUTER VISION AND TARGET TRACKING

The target identification and tracking are the key component to this robotic fish platform. Without a way to accurately detect lionfish, this project has little additional value over existing robotic fish or traditional AUV's. Therefore, it was of paramount importance to choose the most robust tracking algorithm for this application.

4.1 Color Spaces

To understand how images can be manipulated by a computer there are several basic concepts that it is necessary to understand. The first is how image data is stored and represented. When a digital image is captured or an analog image converted to digital, it is sampled and quantized at regular intervals, pixels, and intensity values stored at each pixel. Common image types include 1 sample per point, 3 samples per point, and 4 samples per point. These samples are called channels.

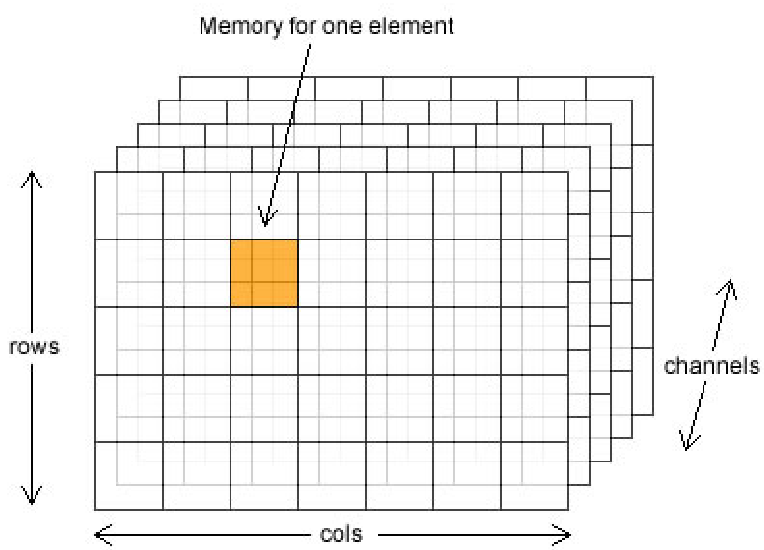

Figure 4.1: A Generic Image Matrix.

Values are stored as eight bit integers, meaning each channel has a value in the range of zero to 255. A single channel image is black and white or grayscale. Images with three channels can be stored in one of several color spaces. The most typical is RBG or Red, Blue, Green space. Each channel stores the intensity of one of these three colors, and when combined can display any of millions of colors. The RBG color space can be represented as a rectangular coordinate system.

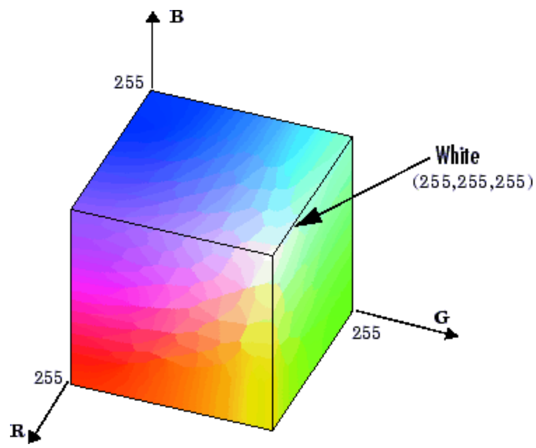

Figure 4.2: RBG Color Space [31].

Another typical color space is HSV, or Hue, Saturation, and Value. This is essentially a coordinate transform into a cylindrical coordinate system. Hue values are in the range from 0-179, Saturation 0-255, and Value 0-255. The range of Hue values is 0 to 180 degrees, because a full 360 degrees is outside the range of an 8 bit integer's ability to capture.

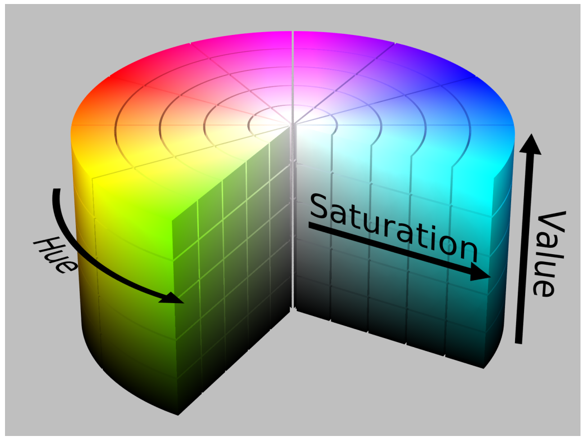

Figure 4.3: HSV Color Space [32].

Images that have 4 channels use the first three channels for color information and the fourth channel for an opacity value. This representation will not be used in this application.

4.2 Tracking Methods

Five different object tracking methods were compared. The five methods chosen were Background Subtraction tracking, Mixture of Gaussian (MoG) adaptive background tracking, Speeded Up Robust Feature (SURF) detection tracking, Color tracking, and CamShift tracking.The following sections will explain the basic underlying theory of each and the various advantages and drawbacks inherent in each method. Each of these methods is then compared and the most robust method for autonomous underwater tracking of lionfish is determined.

4.2.1 Background Subtraction

Background Subtraction tracking works through the simple principle of comparing two subsequent video frames and determining what has changed. The first step is to read in two sequential frames from our video, which it then converts to gray scale. This means that each pixel on of the image has only one channel, with a value between zero and two-hundred and fifty-five. Zero corresponding to black and two-hundred and fifty-five being white.

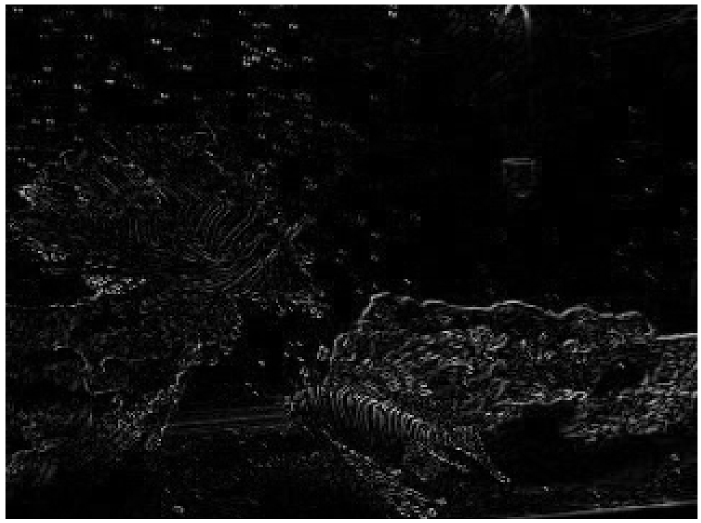

Figure 4.4: The absolute difference between two subsequent frames.

The absolute difference of the two images is then taken. This operation returns the amount that the value of each pixel has changed. If a pixel has changed by a value below a sensitivity value it is set to zero (black), above the sensitivity value to two-hundred and fifty-five (white). This results in a binary color image where the non-moving background (black) has been separated from the changing foreground (white). A blur function can then be performed on this to remove noise. Blur essentially takes a weighted average of the surrounding pixels based on a specified kernel, usually a Gaussian.

Figure 4.5: After Thresholding Operation.

Figure 4.6: Thresholded Image After Blur Operation.

A contour finding function can then be used to find the outlines of each foreground section. It is assumed that the contour encompassing the largest area is the lionfish to be tracked. The bounding rectangle of this largest contour is found and the center of the rectangle used as the output of the tracking program.

Figure 4.7: Background subtraction tracking.

There are several advantages and disadvantages to this method. The main advantage to this method is its simplicity and speed. It is a method that would work well with a stationary camera and slowly changing lighting conditions. However, if the camera is moving, then the background pixels will be constantly changing, making them indistinguishable from the moving object that is actually trying to be tracked. The same problem is encountered with abrupt lighting changes. Slowly moving objects, such as swaying plants or moving clouds also pose a potential problem. If all the leaves on a tree are moving slightly, this method would detect a large object that appears to be in motion. Another problem with this method is that there is no distinguishing between objects, it merely checks for any movement at all. This makes it impossible to track a specific target, if anything else is moving in the frame it will register as a false positive. Additionally, assuming that the largest contour found is the desired object to be tracked is not a robust solution.

4.2.2 Color Based

Another tracking method is based on color. The underlying theory is that it is possible to find a range of colors by specifying upper and lower limits on each channel. First, a frame is read from a video. It is converted from BGR color space to HSV color space. In general, HSV color space has a larger difference in intensity values between colors, which helps to facilitate thresholding and tracking. Minimum and maximum

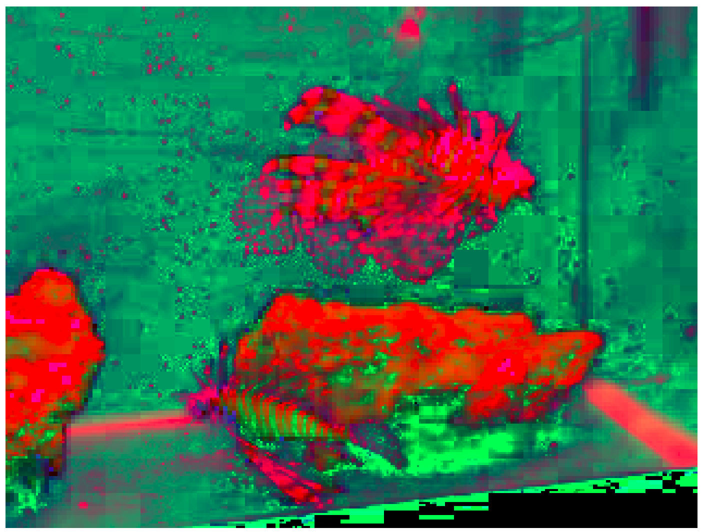

Figure 4.8: Frame in HSV Color Space.

values for hue, saturation and value are experimentally determined depending on the color of the object being tracked. Any pixel with values outside of this range is set to black, and anything inside the range is set to white. The resulting outline contours are computed and the bounding rectangle of the contour encompassing the largest area is found. The center of this bounding rectangle is returned as the point being tracked.

There are several problems with this method. The first is that it is not very discriminating. The range of values for the threshold has to be wide enough to account for lighting changes as the lionfish movies. This means that anything even close to the color of the lionfish is tracked. This makes it very inaccurate when tracking anything outside of a structured environment where it can be ensured that it is the

Figure 4.9: Color Thresholds.

Figure 4.10: Color Tracking.

only object in the scene with that color, such as tracking a tennis ball on a black background. Additionally, like the background subtraction method, assuming that the contour enclosing the largest area is the desired object to be tracked is not robust and leads to many false positives. It also requires manually tuning the minimum and maximum threshold values, which can be time consuming and potentially lead to further inaccuracies.

4.2.3 Mixture of Gaussians(MOG) Background Subtraction

Another method to segment foreground from background images and track motion is to use an adaptive background mixture model, as proposed by Stauffer and Grimson. This is also known as the Mixture of Gaussians background subtraction method, or MOG. There are two main components to this method, a probabilistic model for sperating the background foreground, and a technique to track the foreground object. As a video plays, each pixel changes value as time progresses. This history of values is how we define the pixel process of a pixel X as its value changes from frame 1 to frame t.

$$Pixel\ Process: \quad [X_1 \dots X_i \dots X_t] \quad where\ 1 \leq i \leq t \tag{4.1}$$

This recent history can be modeled by a mixture of Gaussian distributions. The probability of observing a certain value X at frame t can be expressed by the function provided by Stauffer [33].

$$P(X_t) = \sum_{i=1}^{K} \omega(i,t) * GaussianPDF(X_t, \mu(i,t), \Sigma(i,t) \tag{4.2}$$

In which K is the number of Gaussian distributions used in the model, $\omega(i,t)$ the weighting term that describes what fraction of the data is represented by the i^{th} Gaussian in the mixture at time t, $\mu(i,t)$ is the mean value of the i^{th} Guassian in the mixture at time t, and $\Sigma(i,t)$ is the covariance matrix of the i^{th} Guassian in the mixture at time t. The GuassianPDF is a Guassian probability density of function.

In the next frame, there is a new pixel value X_t, where $t = t+1$. The probabilistic model must be updated with this new information. The value X_t is compared to each Guassian in the mixture. If it is less than or equal to 2.5 standard deviations from the mean, then that Gaussian is labeled as matched and not further Guassians are inspected. If it is greater than 2.5 standard deviations, that Gaussian is labeled as unmatched and the program moves on to the next. After all Guassians are labled as matched or unmatched, there are three possible cases [33].

1. If Gaussian is marked as matched: increase the weight, adjust the mean closer to X_t, and decrease the variance.
2. If Gaussian is marked as unmatched: decrease the weight.
3. If all Gaussians in the mixture are marked as unmatched: Mark as a foreground pixel, then find the least probable Gaussian in the mixture and set its mean to X_t, variance to high, and weight to low.

After following this procedure for all pixels, all foreground pixels will be known. It is now necessary to determine which Gaussians from the mixture most accurately represent the background pixels. These are the distributions with high weight and low variance. To find these distributions, divide the weight by the standard deviation for each Gaussian in the mixture. Order the Gaussians from largest to smallest, according to this parameter. In this order, sum the weights until greater than a predetermined threshold. A larger threshold level means more pixel values are included in the background. Represent the background with the means from each Gaussian in the sum [33].

To summarize, a model representing the background and a marked set of pixels representing the foreground are now known. It is now necessary to determine how these foreground pixels are connected and which pixels belong together in the same object. This can be accomplished by a segmentation or connected components method, such as Horns algorithm. [33]

The MoG method can be quite accurate in the proper application. However, it assumes that the camera remains stationary relative to the background, which is not the case for this application. It becomes confused about which objects are in the background and which are in the foreground.

Figure 4.11: Pixels Marked as Foreground.

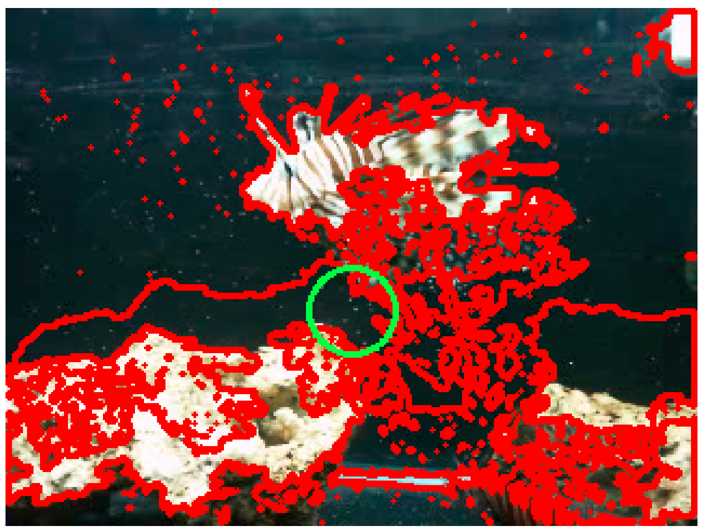

Figure 4.12: MOG Contours and Tracking.

4.2.4 SURF Object Detection

A fourth detection method is based on extracting and matching feature points. This method is uses a method known as SURF, for Speeded Up Robust Features. This is a method for finding corresponding points of interest in two different images [34]. Distinctive interest points such as corners, blobs, and junctions are selected. The region around each interest point is described by a feature vector. This description must be distinctive to the point being described and at the same time resistant to noise, detection errors, and both geometric and photometric deformations. These descriptor vectors are then matched between images. There are different methods of completing this matching, but most often it is based on the distance between feature vectors. SURF provides both a detector and descriptor [34].

The main problem with this method is that it requires a reference image that matches exactly the object in the scene that is being examined. This is because the feature points found in the scene and their description vectors must be the same in the both the reference image and the scene being examined. If they are not, then no matches will be found, or the matches will be incorrect. In a biological setting, meeting this requirement is nearly impossible. This is because the image in the scene is constantly changing orientation and lighting conditions. Adding additional complications is the fact that the object being tracked, the lion fish, is colored to blend in with the rocks. This leads to lots of false matches, which makes it impossible for the tracker to converge.

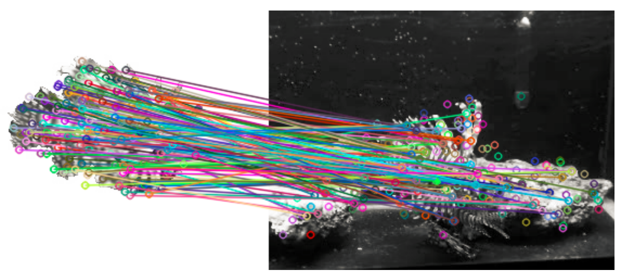

Figure 4.13: SURF Matching Points.

4.2.5 CamShift

The final object tracking method is the CamShift algorithm. It is also a color based tracking method, but is far more robust than the simple color tracking explained previously. CamShift is essentially an extension of the MeanShift algorithm. MeanShift is, at its most basic level, a way to find local peaks in a probability density function. [35] Given a distribution of data points and a search window, the center of the search window is shifted towards the center of mass of the area covered by the window. This process is iterated until the position converges. This finds the region of highest density. [36] If this process is used on a probability density map, it converges to the local region of highest probability.

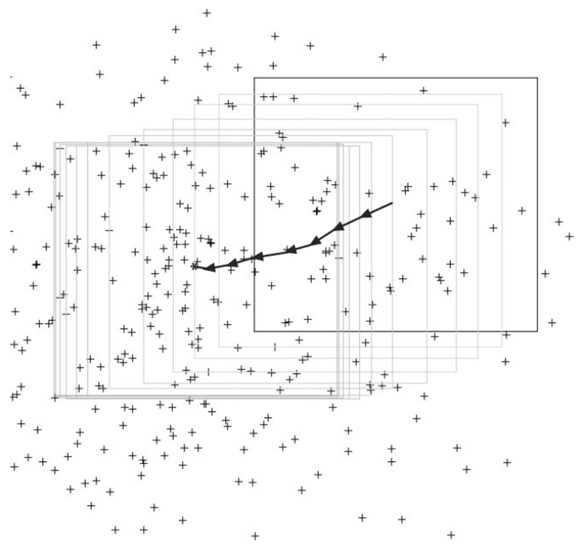

Figure 4.14: MeanShift Tracking.

This process can be used, when combined with some image pre-processing, to locate a specified object in an image or video. The downside of this method is that the size and shape of the search window are immutable. For example, depending on the initial placement of the search window different regions of highest density could be found. Or, the region of highest density may be very small or large compared the search window. To solve these issues, the MeanShift algorithm is extended. Based on the image moments of the detected high probabilty region, the search window can be scaled and rotated. This makes it possible to determine the size and orientation of the found object [37].

For visual tracking, the goal is to compare a reference image to an incoming video stream and locate areas in the video frames that match the reference image. This is accomplished by processing the reference image and video frame separately, and then compare them using CamShift.

The first step in this process was to read in the reference image and convert it to RSV colorspace. In general, this allows for better contrast between colored regions. Next, it is important to determine which portion of the reference image is the actual object to be tracked, not just the background of the image. To do this, a mask was created. Threshold ranges for hue, saturation, and value were experimentally determined. Anything outside this range was set to 0 (black),and anything inside the range to 1 (white).

Figure 4.15: Reference Lion Fish Image.

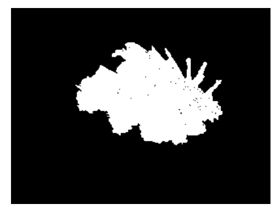

Figure 4.16: Binary Mask Image.

It was then possible to use this binary image to mask the regions of the reference image that aren't relevant. The hue channel was extracted from this masked image. This is the channel that will have the highest contrast between the background and tracked object in most cases. A histogram was then calculated from this hue channel. This histogram is what will be compared to the incoming video stream. A comparison of the histograms from a masked and un-masked image shows the importance of the masking operation, see Fig.4.17. The mask eliminates the excess histogram data from the background and return only the information from the object itself.

In order to maximize the accuracy for the application of tracking a lion fish in a natural environment, 12 reference images from several different fish were used. All of these images were independently masked and their histograms calculated. These twelve histograms were then averaged. This process eliminated errors resulting from variation in individual lionfish coloring and differences in light and shadow conditions.

Figure 4.17: Histograms of Unmasked and Masked Images.

Potentially higher accuracy could be obtained with an increased number of reference images included in the average, but diminishing returns were observed after 12 images.

After the processing on the reference image was complete, a similar process was conducted on the incoming video frames. The frame was converted to HSV, masked, and the hue channel extracted. It was then possible to calculate a back projection map. A back projection map is an image which maps the probability that a given pixel in the video frame corresponds to the reference image. It was then possible to use the CamShift algorithm on this back projection map to find the region of highest probability.

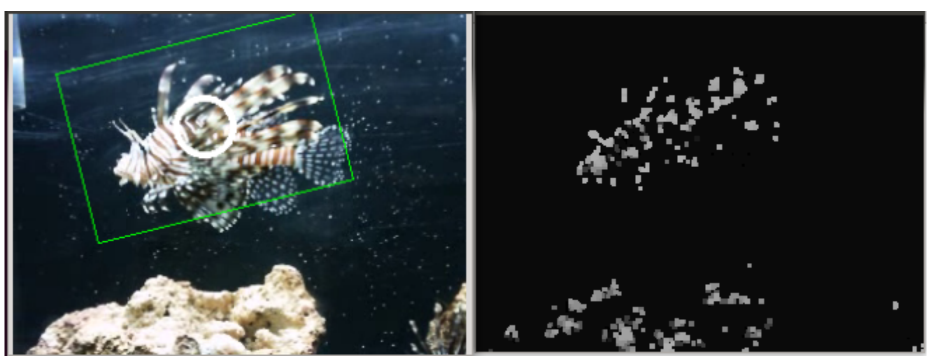

Figure 4.18: Video Frame and Corresponding Back Projection Map.

After the back projection image was calculated, it had a decent amount of noise. This generally took the form of stray pixels or groups of pixels. It is desirable to reduce this noise before feeding the back projection into the CamShift algorithm. This was accomplished in this application by applying the erode and dilate functions, in that order. The erode function scans the image with some kernel, usually a square or circle. As the kernel is scanned over the image,the minimum value being overlapped is found, and the pixel at the center of kernel replaced with this value. Essentially, if a white pixel has a black pixel in its neighbors, the white pixel is replaced with black.

Dilation is the opposite of erosion. As the kernel scans, the maximum value is used to replace the center of the kernel. In practice, if a black pixel has a white pixel in its neighbors, the black pixel is replaced with white. It was necessary to use both of these functions in sequence. In the case of the back projection map, the noisy pixels are white, so erode removed isolated pixels and small blobs. Larger white pixel clusters are also reduced in size, but do not disappear entirely. Dilation restored these close to their original size. It should be noted that this process removes a great deal of detail from the image and tended to "smooth" things out. In this application, the benefits of noise reduction greatly out weight the small negative of reduced detail.

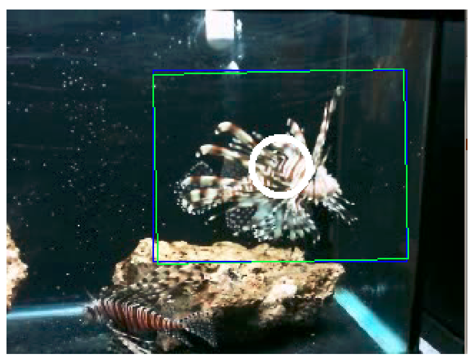

Figure 4.19: Frame that Back Projection is Calculated For.

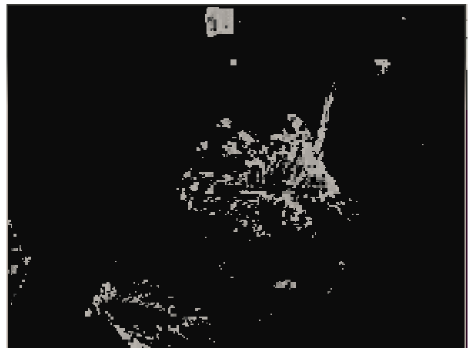

Figure 4.20: Back Projection Map Before Dilation and Erosion.

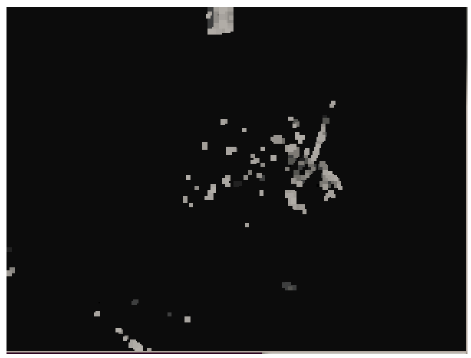

Figure 4.21: Back Projection Map After Dilation and Erosion.

The CamShift algorithm requires three inputs; the back projection image, a reference histogram, and an initial rectangular search window. When specifying the search window there are two options, to use the rectangle bounding the object found in the previous frame, or to reset the search window to its initial position for every frame. Each method has drawbacks and advantages.

The recursive method of using the position and size of the window from the previous frame converged more quickly than if the search window was reset every frame, however, it only searched a fraction of the scene. This maked it more robust to distraction from other objects in the scene and it stayed closely tracking the original object that it found. This is good if there is high confidence in the initial convergence and the target is not occluded or departs the frame. However, if the initial match is incorrect, then the tracker will continue tracking this incorrect object and ignore a correct one.

If the search window was reset after every frame, there was slower convergence as the entire frame had to be searched. However, it found the highest probability region in every frame, without regard to what was previously being tracked. This is good if the target has entered from out of frame or has been occluded. However, it is more likely to track an incorrect object if there are distracting objects in the scene, such as similar colored background object or multiple targets in the window at the same time. In this application, higher accuracy was achieved with resetting the search window after every frame.

In order to reduce false positives, a shape parameter was included. When an object was detected, the search window converged to a scaled and rotated bounding rectangle. If the aspect ratio of this rectangle fell outside of an experimentally determined range, a false positive is returned. This is to avoid such scenarios as when the program includes an adjacent rock or coral formation in the detection window, stretching it into an obviously incorrect shape., as seen in Fig. 4.22.

4.3 Results of Comparision

Each of these six methods was tested on a compilation of video footage I had taken of live lion fish in a simulated reef environment. The test video includes many of the factors that a bio-inspired AUV would encounter in operation such as fish entering and exiting the frame, multiple fish in the frame at one time, fish being occluded by other objects in the scene, background objects similar in color to the fish, and relative motion between the camera and the background. Each method was examined by running through the test footage frame-by-frame and assigning a YES or NO value to each frame. A YES value designated that the position of the lionfish was accurately determined in that frame, a NO value if the position was inaccurate or not returned at all.

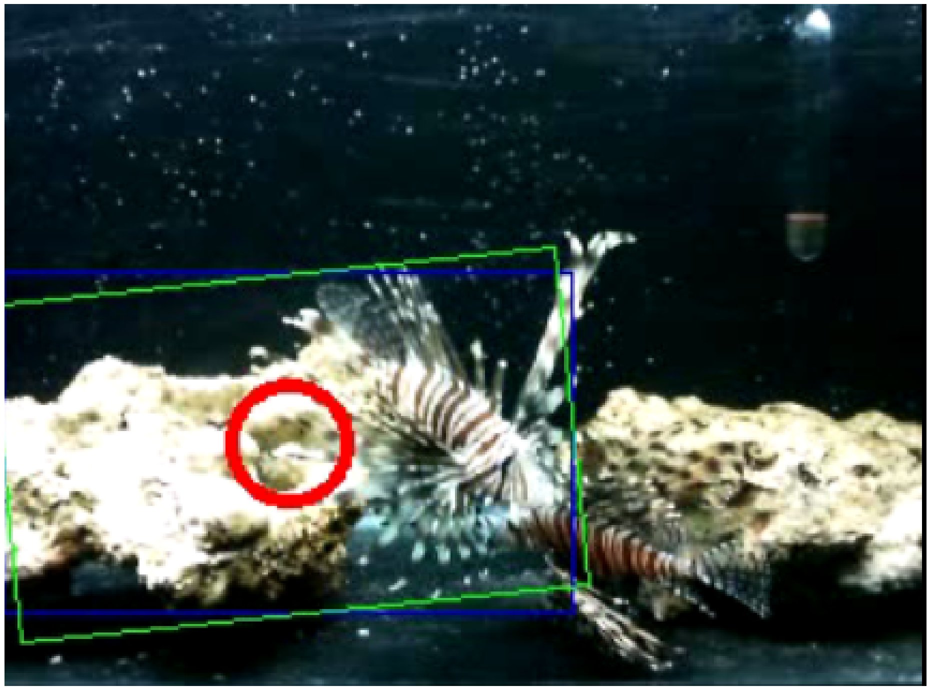

Figure 4.22: A false positive identification.

Table 4.1: Comparison of Tracking Methods.

Name	Yes	No	Frames	Percent
Background Subtraction	457	543	1000	45.7
Color	38	1962	2000	1.9
MOG	360	1640	2000	18
SURF	0	2000	2000	0
CamShift	1479	521	2000	73.95

It can be seen that the CamShift based tracking is the most accurate by a considerable margin. This demonstrates that for an autonomous underwater vehicle in an unstructured and unregulated environment, the most robust of the compared method of target identification and tracking is the proposed CamShift algorithm. There are several reasons for this outcome. Both the backgrond subtraction and MOG methods are based on the assumption that the camera stays stationary with respect to the background. In this application the movement of the camera is fast enough that this is not a valid assumption. The color tracking method is similar to the CamShift

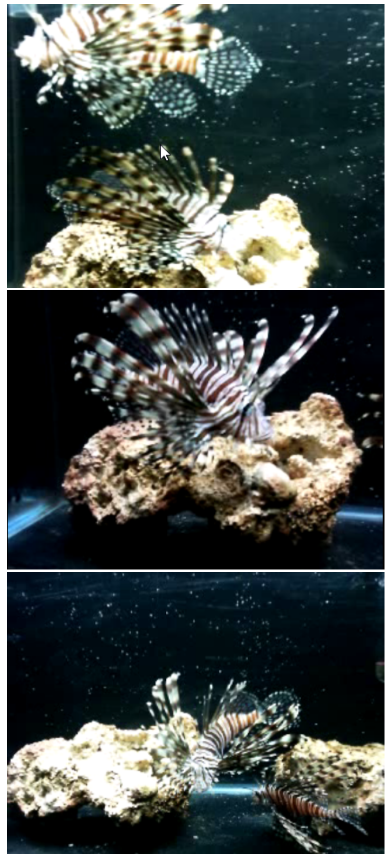

Figure 4.23: Selection of frames from test video.

method in that both are dependant on color information. However, where it is fails is that after the thresholding and contour operation, there is no way to distinguish which contour is most closely resembles the reference object. CamShift solves this with the use of the back projection map. SURF feature detection turns out to be not useful in this application. The feature points on the video images cannot be matched to the reference image. This is because the shape, orientation, and lighting of the lionfish are constantly changing and are never the same as in the reference image. This detection method works well for rigid objects, but not for living creatures.

5. PRELIMINARY CONTROL AND VALIDATION

Preliminary design of a control algorithm and validation of the robotic platform were performed. The overall performance of the robotic fish was tested to ensure correct operation of the actuators, sensors, and on-board performance of the vision system.

5.1 Validation Testing

Several preliminary validation tests were performed. The first was a basic test of the locomotion. The fish was run on purely open loop control in a small pool. This demonstrated that the fish was able to produce sufficient thrust and that no leaks occured during normal operation. Previous to this, tests had only been performed in an oil tank, while tethered.

After it was confirmed that forward locomotion and turning maneuvers were possible, the vision system was tested. This was to ensure that the system was capable of functioning while under full operation load. The estimated power requirements were within the range of what the battery system was capable of proving, but it was important to verify that full image processing capabilities remained intact while the actuators and other power hungry systems were operating.

This test was conducted by dividing the video frame into four equally sized columns. If the target, a tennis ball was used for this test, was detected in the left most column, the right pectoral fin would beat at 5 Hz and the left would beat at 2 Hz. This results in a sharp turn to the left, moving to center the ball in the frame. If the ball was detected in the inner left column, the right pectoral fin would beat at 3 Hz and the left at 2 Hz, resulting in a slower left turn. The opposite happened if the ball was detected in the right two columns. This test was conducted in air in order to accurately place the target in the camera's field of view. Results were as expected and confirmed the correct functioning of the vision system.

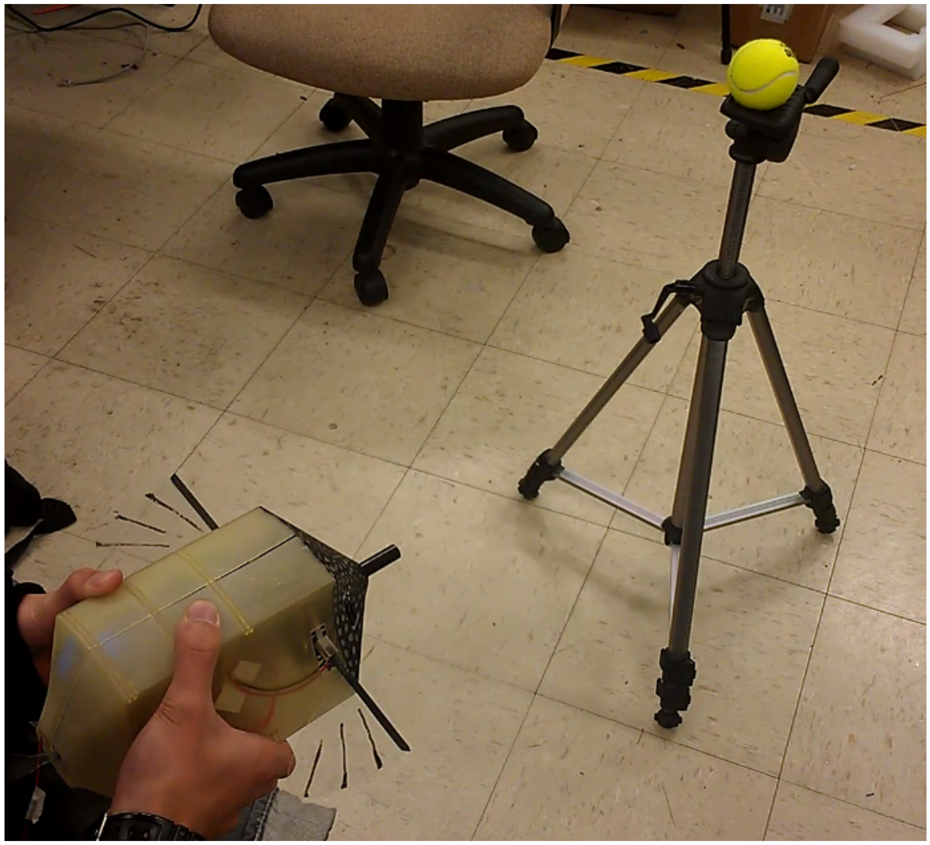

Figure 5.1: Conducting the validation of the vision system. The boxfish like outer shell is visible.

5.2 Preliminary Data for Control System

The design of a feedback control system is the next step in this project. The final goal is for fully autonomous visual tracking of lionfish. The first step towards this is to determine the kinematics of the system. Kinematics data and force data can be mapped to each other and a tranfer function obtained.

The control inputs to the system are the frequency at which the actuators flap the fins. Each pectoral fin and the tail fin are actuated separately. The outputs tracked are velocity in the plane and the angular velocity in the yaw direction. Several experiments were run to obtain these velocities across a wide range of input frequencies.

5.2.1 Kinematics Data

The experimental setup consisted of an overhead camera above a pool with a 10 cm grid drawn on the bottom. 17 frequency combinations were input and the x and ya velocities were recorded along with roll, pitch, and yaw rates from the gyroscope and acceleration in three axis from the accelerometer.

Table 5.1: Frequency Combinations.

Right Pectoral(Hz)	Left Pectoral(Hz)	Tail(Hz)
5	5	0
4	4	0
3	3	0
2	2	0
4	5	0
3	5	0
2	5	0
1	5	0
5	4	0
5	3	0
5	2	0
5	1	0
0	0	5
0	0	4
0	0	3
0	0	2
0	0	1

All data from the IMU was recorded directly to the mbed. To obtain X and Y position data, overhead video was recored while this test was conducted. A modified version of the CamShift lionfish tracking program was used to post-process this data. The X and Y coordinates of the robot in the video frame were extracted using this program.

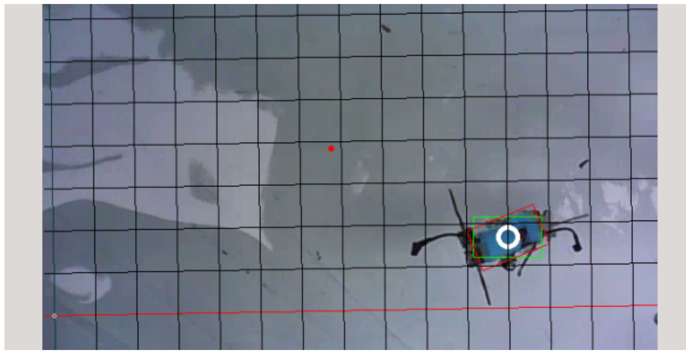

Figure 5.2: The overhead camera tracking program.

A coordinate transform matrix was used to translate, rotate, and scale the video coordinate frame to the frame of the grid on the bottom of the pool.

$$P' = (R * P)^T + S$$

Where R is the rotation matrix, P is a vector of the coordinates in the camera frame, S is the translation of the origin point, and P' is the vector of coordinates in the pool coordinate system. This coordinate transform can be seen in Fig. 5.3 and Fig. 5.4.

$$R = \begin{bmatrix} .999 & .016 & 0 \\ .016 & -.999 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{5.1}$$

$$S = \begin{bmatrix} 12.2258 \\ 324 \\ 0 \end{bmatrix}$$

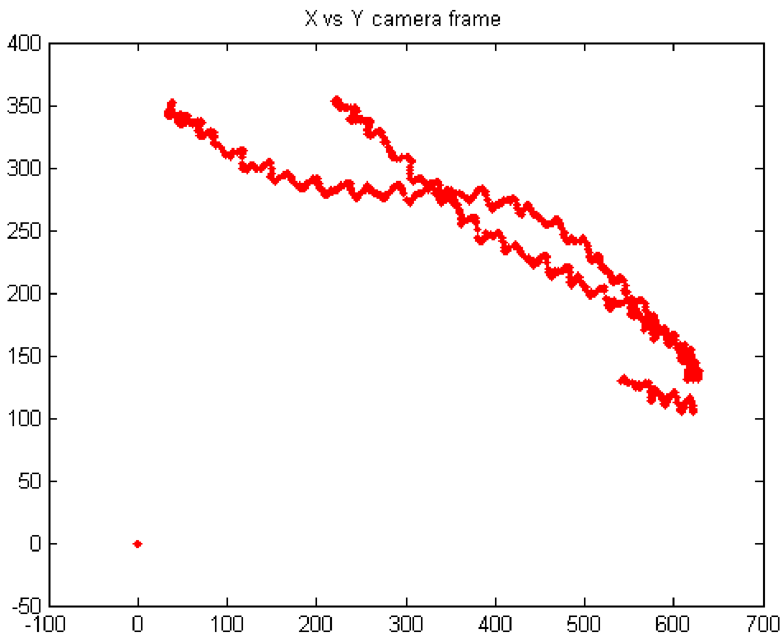

Figure 5.3: The X vs Y position as seen from the camera coordinate system.

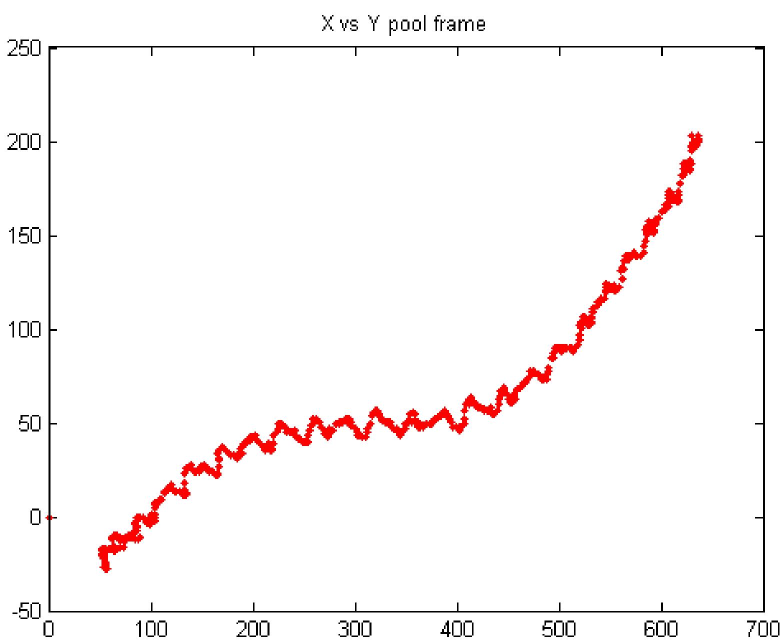

Figure 5.4: The X vs Y position as seen from the pool coordinate system.

5.2.2 Force Data

Force data was obtained for the same set of frequency Combinations. The neutrally bouyant fish was connected to a six axis force sensor in line with the fish's center of gravity using a 3D printed mounting bracket, as seen in Fig. 5.5.

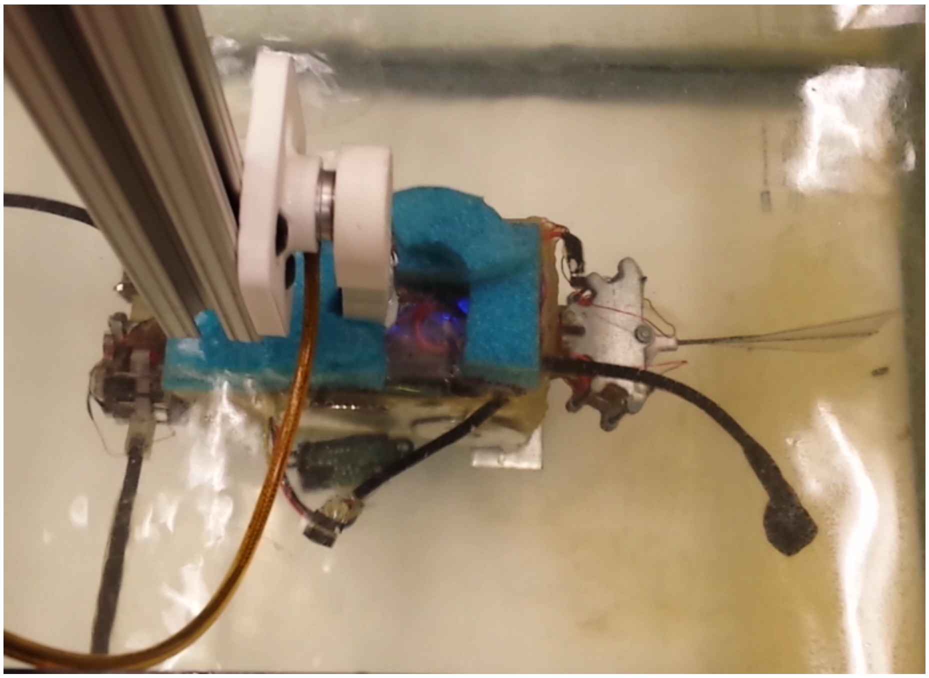

Figure 5.5: Force testing experimental setup.

6. FUTURE WORK

The results from found so far on this project suggest several directions for future work. The most interesting of these are pcb miniaturization and a stereo vision system.

6.1 PCB Miniaturization

The current PCB design is large and has unused features such as connections for a hydrophone. In fact, there is an entire 12 volt power circuit that is unused. It should also be possible to integrate components into the PCB that are currently discreet boards. This includes the mbed processor, inertial measurement sensor, and motor controllers. This would greatly reduce the footprint of the electronics components and allow the entire robot to be downsized.

Using surface mounted components does have some disadvantages, it makes it much more difficult to service the components if something goes wrong. It also locks in the design and makes changes impossible without printing a new board.

6.2 Stereo Vision System

The direction forward with the vision system is a stereo vision system. Stereo vision is composed of two calibrated cameras set a fixed distance apart. This system allows for depth measurements that a single camera, monocular system, is not capable of. This would enable more accurate lion identification and more robust following. The robot would be able to maintain a certain distance from the target and be able to measure the size of the lionfish being tracked.

The main obstacle that would have to be overcome to successfully implement a stereo vision system is the limitations of the Beaglebone Black. Firstly, it has only one usb port to connect a camera to. Using a usb hub is not possible due to bandwidth restrictions. It may be possible to connect a second camera using a different connection, such as ethernet, but this would be a difficult task. Additionally, even if two cameras were successfully connected, the board may not have enough processing power to perform stereo calculations at a fast enough speed to be useful on a moving robot.

LIST OF REFERENCES

LIST OF REFERENCES

[1] P.E. Whitfield, T. Gardner, and S.P. Vives. Biological invasion of the Indo-Pacific lionfish Pterois volitans along the Atlantic coast of North America. *Marine Ecology ...*, 235(Baltz 1991):289–297, 2002.

[2] P. E. Whitfield, J. A. Hare, A. W. David, S. L. Harter, R. C. Muñoz, and C. M. Addison. Abundance estimates of the Indo-Pacific lionfish Pterois volitans/miles complex in the Western North Atlantic. *Biological Invasions*, 9(1):53–64, June 2006.

[3] S. J. Green and I. M. Côté. Record densities of Indo-Pacific lionfish on Bahamian coral reefs. *Coral Reefs*, 28(1):107–107, November 2008.

[4] M.A. Albins and M.A. Hixon. Invasive Indo-Pacific lionfish Pterois volitans reduce recruitment of Atlantic coral-reef fishes. *Marine Ecology Progress Series*, 367:233–238, September 2008.

[5] A. Vasenin. CC BY-SA 3.0, 2010.

[6] R. Clement, M. Dunbabin, and G. Wyeth. Toward robust image detection of crown-of-thorns starfish for autonomous population monitoring. *Australasian Conference on Robotics and Automation 2005*, 2005.

[7] P.R. Bandyopadhyay. Trends in biorobotic autonomous undersea vehicles. *Oceanic Engineering, IEEE Journal of*, 30(1):109–139, 2005.

[8] G. Barbera. Analisi teorica e sperimentale di un sistema di controllo per un veicolo biomimetico Boxfish , 2009.

[9] C.C. Lindsey. Form, function, and locomotory habits in fish. *Fish physiology*, 7, 1978.

[10] M. Sfakiotakis, D.M. Lane, and J.B.C. Davies. Review of Fish Swimming Modes for Aquatic Locomotion. *Oceanic Engineering, IEEE Journal of*, 24(2):237–252, 1999.

[11] J.E. Colgate. Mechanics and control of swimming: a review. *Oceanic Engineering, IEEE Journal of*, 29(3):660–673, 2004.

[12] J. R. Hove, L. M. O'Bryan, M. S. Gordon, P. W. Webb, and D. Weihs. Boxfishes (Teleostei: Ostraciidae) as a model system for fishes swimming with many fins: kinematics. *The Journal of experimental biology*, 204(Pt 8):1459–71, April 2001.

[13] I.K. Bartol, M. Gharib, D. Weihs, P.W. Webb, J.R. Hove, and M.S. Gordon. Hydrodynamic stability of swimming in ostraciid fishes: role of the carapace in the smooth trunkfish Lactophrys triqueter (Teleostei: Ostraciidae). *Journal of Experimental Biology*, (206):725–744, 2003.

[14] I.K. Bartol, M. Gharib, P.W. Webb, D. Weihs, and M.S. Gordon. Body-induced vortical flows: a common mechanism for self-corrective trimming control in boxfishes., January 2005.

[15] D. Weihs. Stability versus maneuverability in aquatic locomotion. *Integrative and comparative biology*, 42(1):127–34, February 2002.

[16] I.K. Bartol, M.S. Gordon, P. Webb, D. Weihs, and M. Gharib. Evidence of self-correcting spiral flows in swimming boxfishes. *Bioinspiration & biomimetics*, 3:014001, March 2008.

[17] B.L. Partridge and T.J. Pitcher. The sensory basis of fish schools: relative roles of lateral line and vision. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 135(4):315–325, 1980.

[18] C.C. Hemmings. Olfaction and vision in fish schooling. *Journal of Experimental Biology*, 45(3):449, 1966.

[19] N.G. Hairston, K.T. Li, and S.S. Easter. Fish vision and the detection of planktonic prey. *Science*, 218(4578):1240, 1982.

[20] C.W. Hawryshyn and W.N. McFarland. Cone photoreceptor mechanisms and the detection of polarized light in fish. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 160(4):459–465, 1987.

[21] U. E. Siebeck and N. J. Marshall. Ocular media transmission of coral reef fish–can coral reef fish see ultraviolet light? *Vision research*, 41(2):133–49, January 2001.

[22] C. Hawryshyn. Polarization vision in fish. *American Scientist*, 80(2):164–175, 1992.

[23] M. Dunbabin, J. Roberts, K. Usher, G. Winstanley, and P. Corke. A Hybrid AUV Design for Shallow Water Reef Navigation. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, (April):2105–2110, 2005.

[24] M. Dunbabin and J. Roberts. A new robot for environmental monitoring on the Great Barrier Reef. *... Conference on Robotics ...*, 2004.

[25] M. Dunbabin, P. Corke, and G. Buskey. Low-cost vision-based AUV guidance system for reef navigation. *ICRA 2004*, (April):7–12, 2004.

[26] P. Kodati, J. Hinkle, and X. Deng. Micro autonomous robotic ostraciiform (MARCO): hydrodynamics, design and fabrication. *IEEE Transactions on Robotics*, 2008.

[27] P. Kodati. *Biomimetic micro underwater vehicle with ostraciiform locomotion: System design, analysis and experiments*. PhD thesis, 2007.

[28] P. Kodati. *BIOMIMETIC MICRO UNDERWATER VEHICLE WITH OSTRACIIFORM LOCOMOTION : SYSTEM DESIGN , ANALYSIS AND EXPERIMENTS*. PhD thesis, 2006.

[29] Y. Hu, W. Zhao, G. Xie, and L. Wang. Development and target following of vision-based autonomous robotic fish. *Robotica*, 27(07):1075, March 2009.

[30] C. Spampinato, Y. Chen-Burger, and G. Nadarajan. Detecting, tracking and counting fish in low quality unconstrained underwater videos. *on Computer Vision*, 2008.

[31] V. Golob. Naloga: 10.7: Difuzija napaka. http://www2.lecad.si/vaje/resitve/10.7/index.html, 2004.

[32] Wikimedia Commons. CC BY-SA 3.0, 2010.

[33] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, pages 246–252.

[34] H. Bay, T. Tuytelaars, and L.V. Gool. SURF : Speeded Up Robust Features.

[35] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[36] R. Collins. Mean-shift Tracking Appearance-Based Tracking. 2006.

[37] D. Exner, E. Bruns, D. Kurz, A. Grundhofer, and O. Bimber. Fast and reliable CAMShift tracking. 2005.