Autonomous tracking of mussels in a lab environment

Mehmed Bilal Diken
University of Iowa

AUTONOMOUS TRACKING OF MUSSELS IN A LAB ENVIRONMENT

by

Mehmed Bilal Diken

A thesis submitted in partial fulfillment
of the requirements for the Master of
Science degree in Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

May 2012

Thesis Supervisor: Professor Anton Kruger

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

____

MASTER'S THESIS

____

This is to certify that the Master's thesis of

Mehmed Bilal Diken

has been approved by the Examining Committee
for the thesis requirement for the Master of Engineering
degree in Electrical and Computer Engineering at the May 2012 graduation.

Thesis Committee: ___________________________________
Anton Kruger, Thesis Supervisor

____
Craig Just

____
Guadalupe M Canahuate

To my parents, who strived to their best ability to provide my siblings and I with the best possible education and support.

If I find 10,000 ways something won't work, I haven't failed. I am not discouraged, because every wrong attempt discarded is another step forward.

Thomas A. Edison

ACKNOWLEDGMENTS

I wish to express my appreciation to the Department of Electrical Engineering for their extended long term support, especially Professor Anton Kruger for his guidance and vast reserve of patience and Cathy Kern. I would like to thank Jonathan Durst for taking extra time to help me with fabrication of the physical set up.

I would also like to thank my Thesis Committee: Professor Craig Just and Associate Professor Guadalupe M. Canahuate.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Summary Of Project

Rapid global industrialization and increase in human population over the last century has exponentially increased the demand of fossil fuels for energy, (generation of electricity, fuel for transportation, etc.), and an ever expanding list of fossil fuel derived chemicals, (synthetic fertilizers, pesticides, polymers, etc.), are being used in all aspects of daily life. All of this has inevitably introduced large amounts of nitrogen to earth's Nitrogen Cycle. Thus, one of the challenges put forth by the National Academy of Engineering (NAE) has been the management of the Nitrogen Cycle. (NAE, 2008). Although this challenge is not in the public eye as much as the concerns of CO2 emissions from fossil fuels and its effects on Global Warming, it's anticipated that the effects of human-induced changes to the global Nitrogen Cycle will be profound, and needs to be better studied and understood.

Leaching from agricultural farms, Concentrated Animal Feeding Operations (CAFOs) and runoff to rivers and waterways is known to be one of the most common forms of excess nitrogen being introduced to the environment. A documented case of this condition and its effects is the Gulf of Mexico. Nitrogen delivery to the Gulf from the Midwest via the Mississippi River has caused the hypoxic dead zone to unprecedented levels (Rabalais & Turner, 2001). High nitrogen content in water causes the phytoplankton to multiply exponentially, bacteria and other organisms that feed on phytoplankton than feed and in the process consume most of the dissolved oxygen. The oxygen starved water cannot sustain life, damaging the surrounding fragile ecosystem and killing fish and other organisms. Areas of this nature have been labeled as "dead zones"; the Gulf has pockets that can cover thousands of square miles.

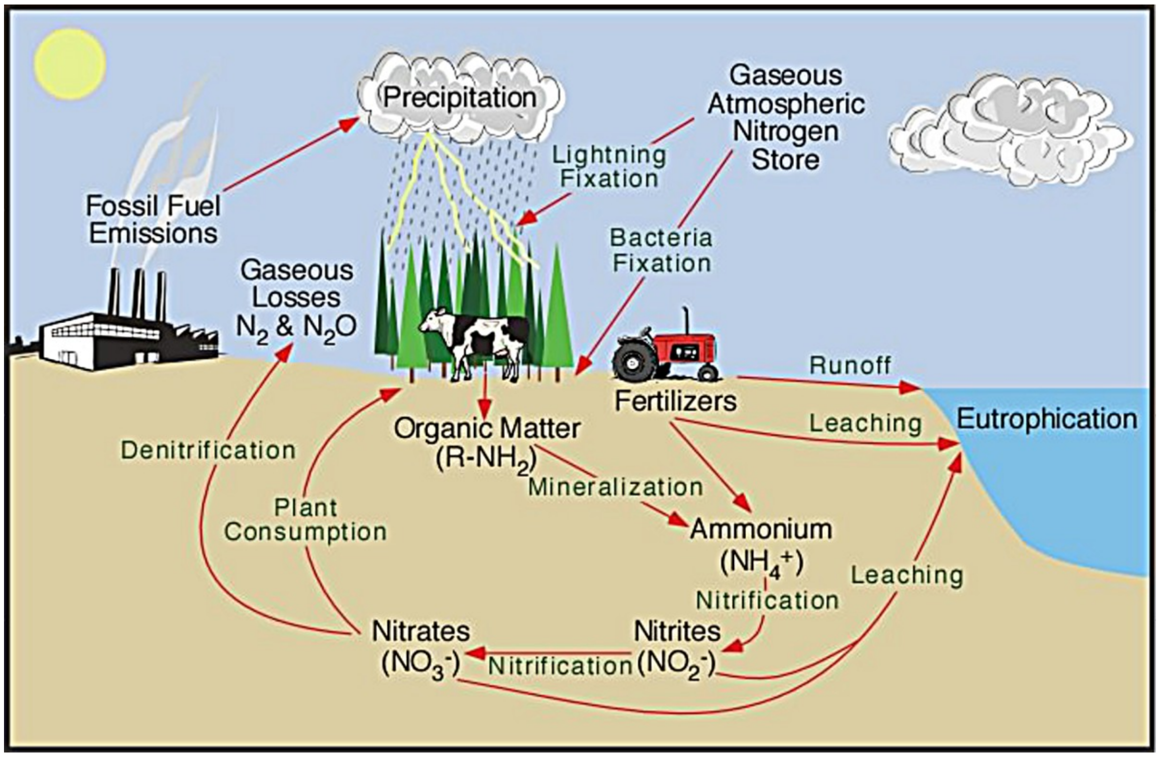


Figure 1 – Nitrogen Cycle. (Pidwirny, 2006)

Mitigation plans for Dead zone management needs better understanding of nitrogen levels in rivers via long term monitoring, modeling and nitrogen delivery to the Mississippi River (Turner, 2008). Currently a major roadblock for environmental scientists is that the current instrumentation and measuring techniques are inadequate to monitor nitrate levels in rivers. Excessive bio-fouling on instrumentation in short period of time inhibits large scale long-term experiments to collect high resolution data on nitrate levels and other elements in river water.

To tackle this problem the goal of the scientists is to build a mussel-based bio-sensory network. The initial prototypes are planned to be launched at the Mississippi River. The live bio-sensing network of mussels will collect nitrate levels along with other relevant water quality data. Servers located at the IIHR—Hydroscience and Engineering at The University of Iowa main campus will process the data which will be used in computer models to integrate and transform the data as a means to further scientists understanding of nitrogen dynamics in rivers.

Previous investigations on mussels were conducted in artificial conditions, mostly in a small scale where the mussels were restricted and tethered. These studies were conducted to mature and test technologies for the possibility of developing systems to monitor mussel un-tethered/wirelessly. The wireless communication between mussels introduces electronics that needs to be mounted external to the shell of these animals. The "big picture" goal of the entire study is to enable scientist to monitor mussels un-tethered in their natural environment.

To achieve this goal we must first verify the following assumption:
"The attachment of sensors and a small "backpack" containing wireless communicators and sensing electronics will have little or no impact on mussel mobility and survival."

Previous experience and anecdotal evidence suggest that large, healthy, adult mussels do not particularly care if a small object attaches itself to the outer shell. There are parasitic mussels that attach themselves sometimes in large numbers on the outer shell of healthy functioning mussels. On the other hand, if the object is too large and heavy, this can clearly affect mussel behavior. What is desirable is that these backpacks should have only a modest effect on short-term survival of mussels, and minimally affect their horizontal and vertical movement behavior.

To test this hypothesis, a small laboratory- scale mussel mesohabitat will be used. Adult mussels will be sourced from the surrounding rivers like Iowa and/or Mississippi river. To enable controlled studies, the mesohabitat consists of two main sections, imaging and tracking setup and bio/chemistry setup. The imaging and tracking mesohabitat will contain de-chlorinated influent water shared between two tanks, periodically spiked with a controlled dose of phytoplankton as a food source. In an effort to mimic the habitat the setup will include:

- a layer of sand to support mussel burrowing and proper feeding orientation
- equipped with time-controlled solar simulators to replicate the diurnal patterns of daylight and darkness experienced by mussels in the river

We will use time-lapse photography to track mussel movements, and digital imagery will be collected and processed via a data-logger and associated computer.

A predetermined number of mussels are selected as the sample and two sets, one with and another without backpacks are placed into each of the mussel-containing chambers. The mussels are of varying sizes, and effort is taken to match the distribution of mussels within the two tanks. Variety of water chemistry disturbances are utilized by scientists to elicit mussel mobility responses, which we monitor with the time-lapse photography. Data is collected on mobility and survivability for periods of time (30 days, 90 days, and 180 days) and data is analyzed for statistical significance.

In the first phase, dummy backpacks are used, similar in size and weight to backpacks with radios and electronics, on the mussels. The time-lapse photography provides information on the mussels' movements, which is crucial for testing this hypothesis. If need be, the data gathered from mussels' movement can be used for consideration when algorithms for wireless-communication are being developed.

Our Project

Tracking movement behavior of mussels in particular is something that is not well documented. We know they move, some species more horizontal than vertical, but the timescale that this happens is not well defined. Is there a trigger for movement like depth of water, or flow rate? How fast do they move to desired location? A day, maybe a couple days? Is it a short bust movement, or it takes them multiple stops? What we know is that they are mostly stationary animals, so it's unpractical to manually watch and observe the behaviors of a tank full of mussels.

In addition to this, the setup for the whole experiment is located in the Water Treatment Plant on the campus of the University of Iowa. This is necessary to have access to river water to best mimic the habitat throughout the seasons in a laboratory environment. Due to increasing security concerns and restrictions over the years having access to an infrastructure building can be challenging. All these factors require the need for an autonomous system that, without human intervention, can detect, track movements of mussels and log them and make this data available to remote users. This involves development of a system that requires multiple questions to be answered and bring together an elegant and scalable solution that the scientist can utilize. Currently there is no plug-and-play solution available to such a problem, and this was the goal of this project.

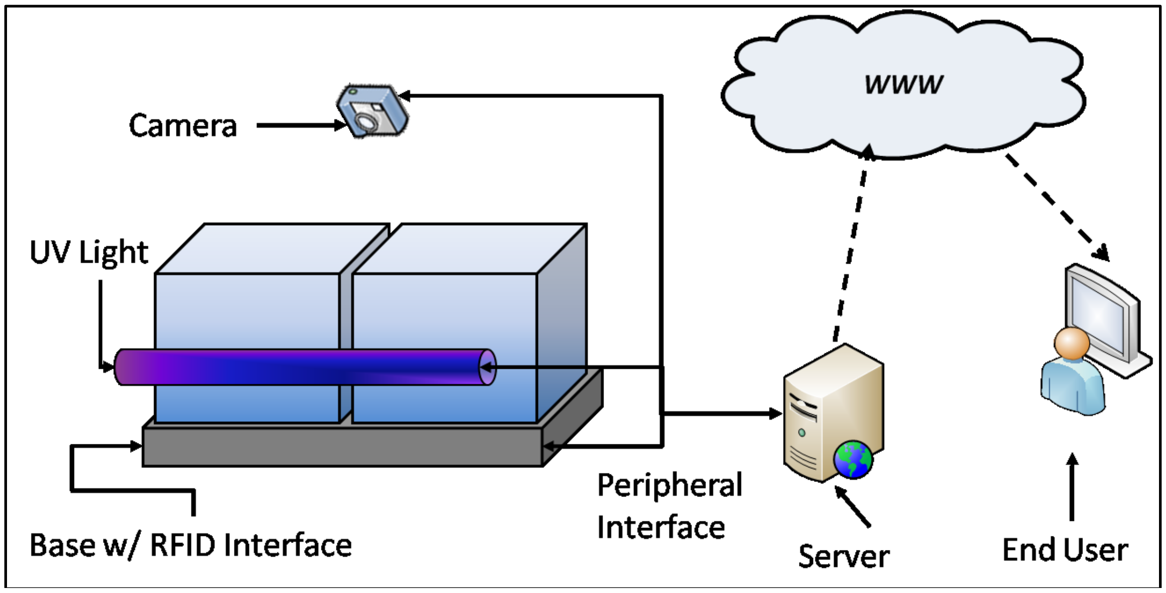


Figure 2 – System Overview: Outline Of All The Main Components Of The System.

Figure 2 above shows the main components of the mesohabitat. Out setup includes a camera suspended above the two tanks, UV lights surrounding the three corners for night acquisition, x/y traverse under the tanks for RFID detection, a webserver to store all extracted information and make this available to the end user. All of these components are external to the tanks to minimize interference with the mesohabitat inside the tanks.

There are challenges to such a task like detection of mussels in water. As described latter sections we encounter detection problems because the shells get covered in bio-foul quickly and they blend with the sediment background. The reflection of light on the surface of the water causes overexposure on the foreground and the mussels underwater could not be detected, challenges of detection at night or during times that there was high concentrations of algae in the water for feeding, etc.

We have looked into multiple scenarios like unique barcodes for each mussel for identification purposes; ultra-violet (UV) UV-illuminated barcodes, painting the shells of mussels with UV activated paint, etc. Each of these methods for identification has their advantages and disadvantages and some of them are not suitable for our needs. We discuss these in detail in the upcoming chapters.

There were challenges with detection of mussels at night, since we are mimicking normal light cycles throughout the tests we needed to develop methods for reliable detection of mussels underwater at night. We explored a variety of techniques including active illumination via LEDs where each mussel would have an LED on it and periodically blink for it to be detected. By using different colors for the different mussels one would they be able to identify the mussels. We explored the use of periodic “black-light” to energize certain pigments on the shells of each mussel and long exposure techniques were explored to find what suits this solution best.

We explored RFID technology to uniquely identify mussels. RFID is also useful water is too murky for image processing to identify the mussels. However, RFID technology is not well suited since penetrations through different mediums vary. We did field tests on different types of RFID tags and readers to find one that would suit our application. A challenge was avoiding collisions of multiple tags that could be energized by the reader. We had to fine tune the RFID reader gain so that we would not energize too many tags at one time. This is a concern for our experiment because the mussels could be clustered very close during moving or when certain undiscovered testing conditions. We share our findings on depth of sediment and water penetration and others in detail in its respective chapter.

The arrangement of this thesis is as follows. The thesis is split into two main sections, namely *Techniques* and *Technologies*. The first few chapters discuss the technologies and techniques employed to solve the given problem these are, image acquisition via a camera, fluorescence, LEDs and RFID technology. Later sections deal with the technologies used to exercise these techniques and bring together all the components to form a functioning solution.

We acquired an API (Application Programmer Interface) for controlling Cannon SLRs, and used it with our Canon 50D SLR camera to develop an application that can control the camera exposure, aperture, remote focusing, and trigger along with other features like live view that are accessible through the controlling computer. The camera is not accessible when hanging above the mussel containing tanks and needs to give the operator the capability to focus, set timer for time-lapse, change aperture, ISO and exposure times before beginning a test cycle.

We used OpenCV and FIJI (open source image processing packages) to process sequential images to detect each mussel and find its location and provide this information to a different layer so it can be logged in a database. Developing an application that can achieve this was challenging and required developing robust image processing pipelines that can efficiently detect mussels’ spatial locations in the tanks.

Wrote PYTHON scripts to run multiple applications like “ffmpeg” and “mplayer” to compile daily and cumulative time-lapse videos for scientist to be able to quickly watch what has happened over a long period of time.

We used WAMP (Bourdon) to set it up and designed dynamic content in PHP so that the current data is always displayed without a need for a webpage developer or a student to constantly update pages to make new data available to the end-user. A MySQL database stores all the location data and setup a web server where the individual images, videos and location information are available to research scientists remotely.

IMAGE ACQUISITION

For capturing the images of the mesohabitat we are utilizing a Canon 50D SLR camera (Canon, 2012). The mesohabitat is a small simulated environment that consists of multiple tanks fed with river water, sediment and plumbing that can mimic a natural habitat of the mussels. The camera has a mid-range zoom lens mounted on it fixed at 18 mm. It is placed above both the tanks to capture images at predefined timeframes. At first the idea of a mid-range SLR might seem to be an overkill compared to a simpler device like a webcam. But the environment that we are operating in call for certain features and flexibility that reasonably an SLR can provide. The test environment is above water, this is inheritably a humid area. So it is important that we have a camera that has adequate weather-proofing. We need to be able to manually control the focus of the camera. Most of the simpler cameras do not have the flexibility to manually control the focus of the camera. This is important because we need to focus on what is under the surface of the water and lock the focus at this setting. This requires the user to manually focus the system before starting the experiment. We can do this remotely via the controls provided to the SLR. The auto focus systems simpler cameras usually focus every time a picture is taken. The problem with this approach is that the reflections on the surface of the water or sharp edges on the tank that are the foreground are more likely to be picked as auto focus points(Mancuso & Battiato, 2001). This will cause the objects of interest, the mussels, to be out of focus in most snapshots. In addition, autofocus performance at night time is unreliable for these cameras. At night there cameras rely, heavily on flash photography. High intensity light on water surface will cause the camera to almost always focus on the undesired foreground.

With an SLR, we can pre-focus as desired and let the camera function at this setting throughout the experiment. Pictures below show the difference focusing makes in our experiment SLRs provide more control. These cameras provide the capability to alter not only ISO sensitivity, and simple manipulations of exposure, but also the ability to change the shutter speed, aperture, metering, etc. Being able to manipulate the camera settings, especially at night times when the shutter needs to stay open the longer without changing any of the other properties is crucial for having consistent images to compile for time-lapse video and image processing. Simpler cameras will increase the ISO sensitivity and then open the aperture of the camera in addition to longer exposure times, which increases the amount of noise in the image and the change the depth of field. Further, the physical sensor sizes between the two camera types are considerable. Smaller sensors are more susceptible to noise and underperform in low light situations (Nakamura, 2006).

A point-and-shoot camera comes with fixed lenses that mechanically move when turned on and off and when zooming in and out, mechanical faults for such cameras are not uncommon. Although same could be said about SLR lenses, they are usually built to a higher standard and in cases when it malfunctions, it can easily be replaced without having to change the body of the camera. This is a huge advantage for SLRs, and also for a set up that can be used for a long time, or adopted to another project easily. Keeping all these factors in mind, SLR approach is by far the best choice for our experiment.

The reason behind taking pictures, it is to gain the ability to detect mussels, in their mesohabitat and gain the ability to track them. It would be ideal if simply taking birds-eye view of the set up would suffice for detection. But there are a many factors that complicate the setup. The initial problem is that all the mussels look very similar, so uniquely identifying them and tracking them for prolonged periods of time simply by taking pictures of them un-altered is not plausible. There are known methods for tracking objects by identifying and tracking unique features and also by estimating region of interest. But these methods will not be useful because of such limitations as bio-foul build up on shells, orientation and murky water. Additionally, mussels cluster closely making any sort of edge or unique feature distinction extremely challenging or impossible.

Figure 3 – Canon 50D: The Camera Utilized For The Project.

Night capture, as stated before is another challenge, Mussel shells that were placed in the tanks during the day and one at night. It is clear that at night, with no external aid such as a light, or some sort of method to identify each mussel, it will be nearly impossible to detect the mussels themselves, not even considering the unique identification.

As we will mention in the upcoming chapters, our biggest obstacle in this project is the bio-foul buildup and the suspended algae. It is very simple and easy to detect mussels during the day in clear water. Simple image processing techniques can be used to differentiate the background (the sediment) from the foreground (the mussels), such as background subtraction (Bradski & Kaehler, Image Parts and Segmentation, 2008), thresholding (Gonzales & Woods, Image Segmentation, 2002), etc. But when the water is murky, either due to suspended bio-foul or cycles of high concentration of algae, it becomes extremely difficult to identify and track each mussel. Other image manipulation methods do not suffice as well because there is simply no way we can capture the data (detail) needed to process the images to extract useful information from.

During initial dry-runs to see if we can simply track mussels, we ran some tests where the camera took pictures of just the shell and relied on the unique features naturally on the shell itself, as shown in the image below. It was simple to detect movement and calculate the position of the mussel as shown in the image below via optical flow methods. Collecting a few snapshots of the shell being moved to different locations we were able to process the images. But we had to abandon the idea quick because it proves to be inefficient and lacking when there are multiple mussels present in the tank and they are clustered closely.

Related Work for Object Tracking

Color Thresholding

Usually when previous knowledge of shape or path of motion is known, a parameterized model of an object can be used, this approach is also referred to as model based tracking. One will frequently hear about this approach when tracking of an object such as a ball in a demonstration or sports, tracking cars in traffic or a hand etc.(Buch, Yin, Orwell, Makris, & Velastin, 2009) Since marking on the mussel shells can change due to bio-foul we would want to look into methods that are less dependent on this feature.

One way to eliminate the geometry dependency is via color thresholding as a feature. This approach is robust to partial occlusion (dirt, algae spots) and is also geometry invariant. It is also efficient, but we are not particularly concerned with this property since we are not processing live video but images that have been captured over time and are already readily available. Richard et al. outlines such an approach. Paper titled “Robust Real-Time Tracking of Non-rigid Objects” (Richard Y. D. Xu, 2004) outlines such an approach.

The approach outlined applies color thresholding to separate the foreground from the background. But prior to this step the image is converted from RGB color space to

YUV color space to reduce its sensitivity to brightness and intensity (Gonzales & Woods, Color Image Processing, 2002). The data is stored in arrays per color component referred to as a color space component array (CSCA). An example is shown in Figure 4below (Richard Y. D. Xu, 2004).

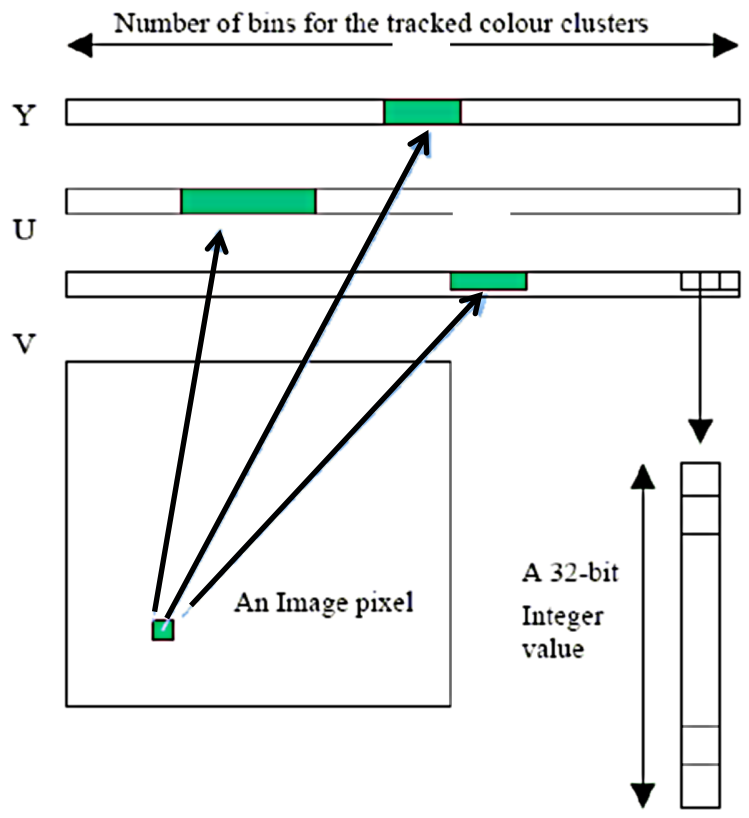


Figure 4 – How An Image Is Stored As YUV Data Representation. (Richard Y. D. Xu, 2004)

Due to how the image is captured through a digital sensor, color is very sensitive to noise; this causes the threshold values to be higher as noise gets higher, so to minimize this noise filtering and region grouping is utilized. First the image is down-sampled using Gaussian blur (Gonzales & Woods, Image Enhancement in the Frequency Domain, 2002) to smooth the image, and then multiple color thresholding is applied. Next a color cluster window filter is applied to each pixel for noise reduction. Region grouping is done via run length coding (Gonzales & Woods, Image Compression, 2002) and a vertical filter is used to remove artifacts from the run length coding (Richard Y. D. Xu, 2004). Figure 5 below show an example of these steps.

Even after clustering to group together all the color clusters of the image together, the region of interest (what we would want to track) also contains color clusters that are in the background, we need to extract the foreground from the background. Clustering is a method that iterates each pixel value in an image that is assigned to a cluster with the closest cluster center which is then updated to be the center mass of all pixel values belonging to that cluster. A simple foreground extraction mask (FEM) is used to achieve this. FEM assigns high positive values towards the center of the selected region and low negative values towards the edge of the region.

Contour extraction (Brandaski & Kaehler, 2008) and Alpha blending is used to make a more robust pipeline that can reliably detect object of interest for tracking purposes. This method can reliably track selected objects that are changing shape due to rotation or orientation of camera or change in lighting due to the using YUV color space that is invariant to light intensity. This application by itself will not work for our project because the user has to manually select the region of interest to track, it is done for a single object at a time, but we can take away useful information on color thresholding methods and background extraction.

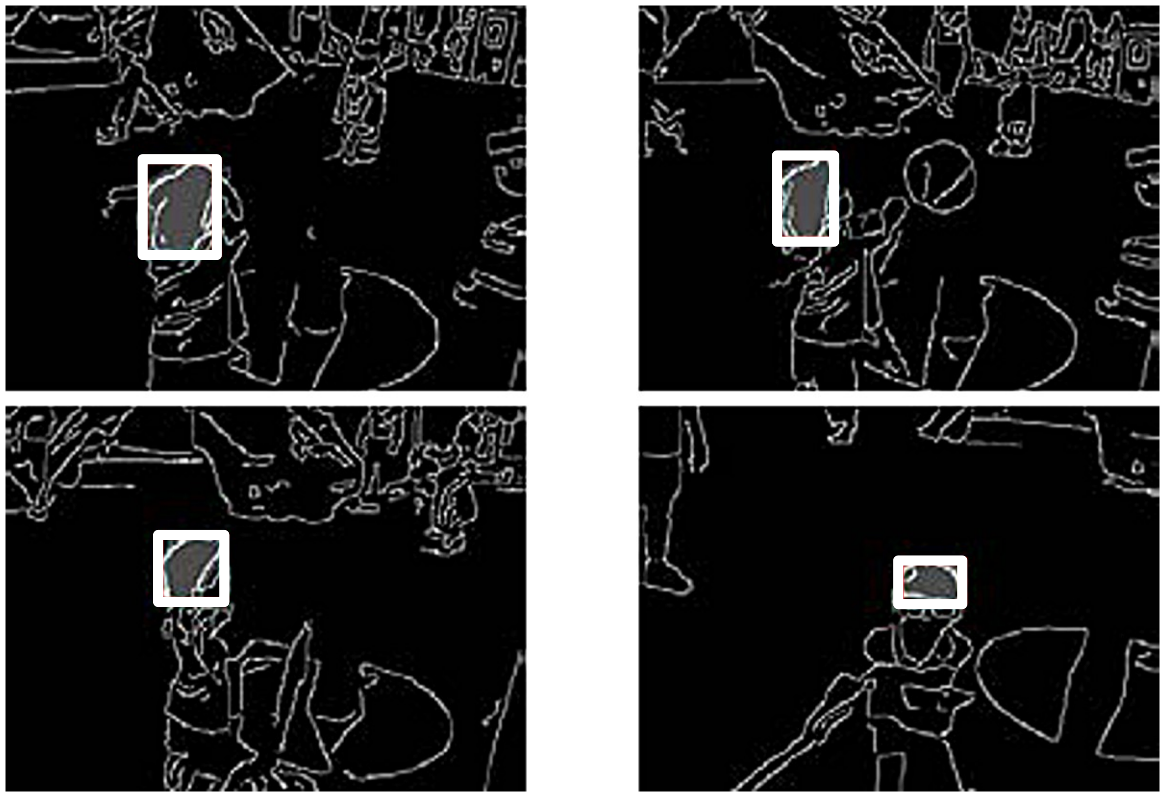


Figure 5 – Tracking Of Object Via Color Thresholding: It Is Clear To See How The Yellow Hat Of The Subject Can Be Tracked Regardless of Changing Shape. (Richard Y. D. Xu, 2004)

Mean Shift Algorithm

Another common method of tracking in computer vision-based applications is the Mean Shift tracking algorithm. The Mean Shift tracking algorithm requires that there are overlaps between the current target searching region and the previous target region (Bradski & Kaehler, Tracking and Motion, September 2008). The Mean Shift algorithm is a nonparametric estimator of density gradient; it will converge to the local maximum probability density function through iteration. The method searches for the target in current frame as the previous target center location to be the center, and move the real target region gradually (Zhou, Pan, & Yang, 2011).

In cases of occlusion or fast moving, the algorithm does not know where to move and match the window; hence it is not able to track properly. Jing Zhou et al. devise a method that can provide better results by introducing a speed and location information to Mean Shift. The pipeline compromises of six steps that can be summarized as follows (Zhou, Pan, & Yang, 2011):

Step1: Initialize target location, speed and weight scale factor

Step 2: Calculate the probability density function of the target template

Step 3: Predict the location of target

Step 4: Execute Mean Shift

Step 5: Determine moving target location; weight of the target location is predicted by the moving information and the target location via Mean Shift

Step 6: Update the target state and return to step 3.

The results obtained from this pipeline is promising for fast moving objects and or deforming objects, but it relies on speed and location prediction, appropriate for maybe a fast-moving object such as a hockey puck but for a mussel that could move around arbitrarily or not move for long periods of time, along with possible durations of high algae bloom which would mean no data to process, this approach would be lacking for adaptation for our project.

Figure 6 below show the selected hockey player that this algorithm can track reliably from a pre-recorded footage. It is able to track the player regardless of the orientation and form that is changing constantly through each frame.

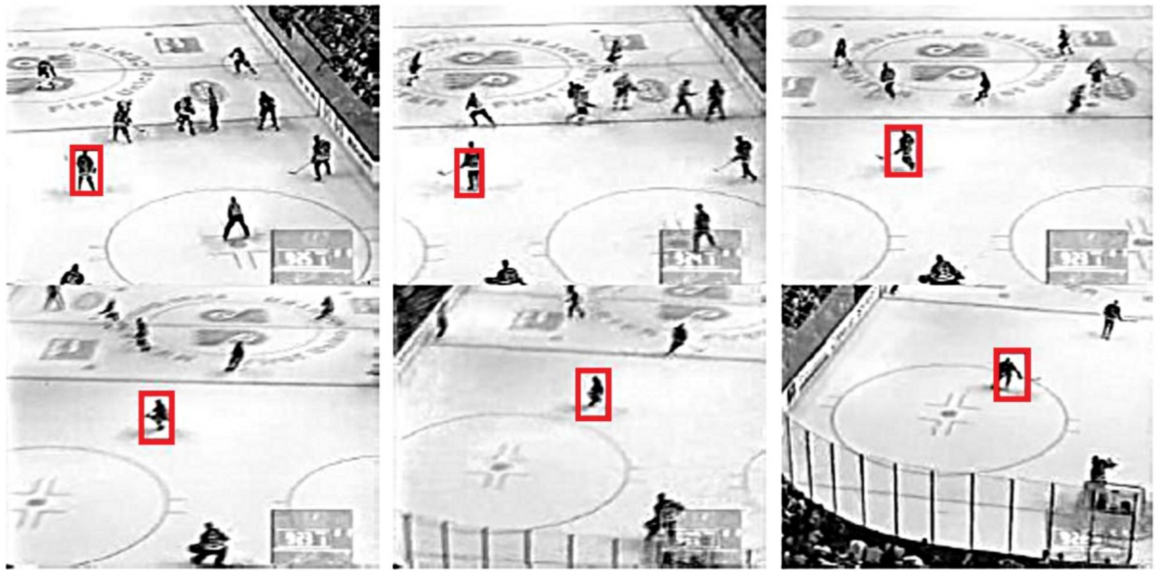


Figure 6 – Tracking With Complex Background: The Hockey Player Being Tracked Can Be Detected Regardless Orientation And Shape. (Zhou, Pan, & Yang, 2011)

Dynamic Feature Graphs

Tracking based on appearance usually model the object to track as a color histogram set (Comaniciu, Ramesh, & Meer, 2003), shape based models use 2D or 3D geometry models such as a gesture or contour based face-and-shoulder detection (Isard & Blake, 1996). In dynamic layer distribution tracking method, a Gaussian distribution of the object pixel values are used (Tao, Sawhney, & Kumar, 2002). All of these methods build appearance models from an initial instance or training sessions and are used to track objects. These methods are successful but do not perform under changing luminance or to objects that can change structure, that be due to movement of the target object or the camera shifting or rotating and even more advanced methods like the shape based tracking that can effectively represent object structure does not take into account the object appearance (Tang & Tao, 2005).

Tang Feng et al (Tang & Tao, 2005) offers a more robust method that can track an object with an evolving feature graph method. Their approach outlines a feature based object representation in the form of attributed regional graph called ARG that they claim to be invariant to various appearance changes. This model is dynamically updated with new features as time passes, so the ARG is always evolving and the introduction of MAP framework for feature based tracking.

To generate the model SIFT features (Lowe, 2004) are used to come up with the primitive basis for the model and are organized into an attributed relational graph. The ARG serves the purpose of encoding the relations between different features, SIFT descriptor is distinct and has high probability to find the exact match under changing illumination and parallel transformations. The feature based tracker proposed by Tang Feng et al. (Tang & Tao, 2005) is formulated into a Maximum A Posteriori, using graph based representation, and the likelihood computation becomes a graph matching algorithm.

The changing object dynamics change over time in the form of a graph in this approach, since there are changing features the graph has to be updated to include new features and remove obsolete features, this is done via tracking features in short-term and long-term basis. New features found on a new frame are not considered mature until they are present in a sequence of frames; only mature long-term features are added or subtracted from the feature graph (Tang & Tao, 2005)

The matching of the current map to the new map is determined through a likelihood function basically sets a relaxation coefficient that allows for the graph to move a little. First step is to build a larger graph than the previous state and use sub graph matching to match the previous model with the graph. Figure 7 below shows an example (Tang & Tao, 2005).

The method for tracking objects outlined in this paper is by far the most robust method that has the capability to track objects that are evolving continuously in comparison to the other methods outlined.

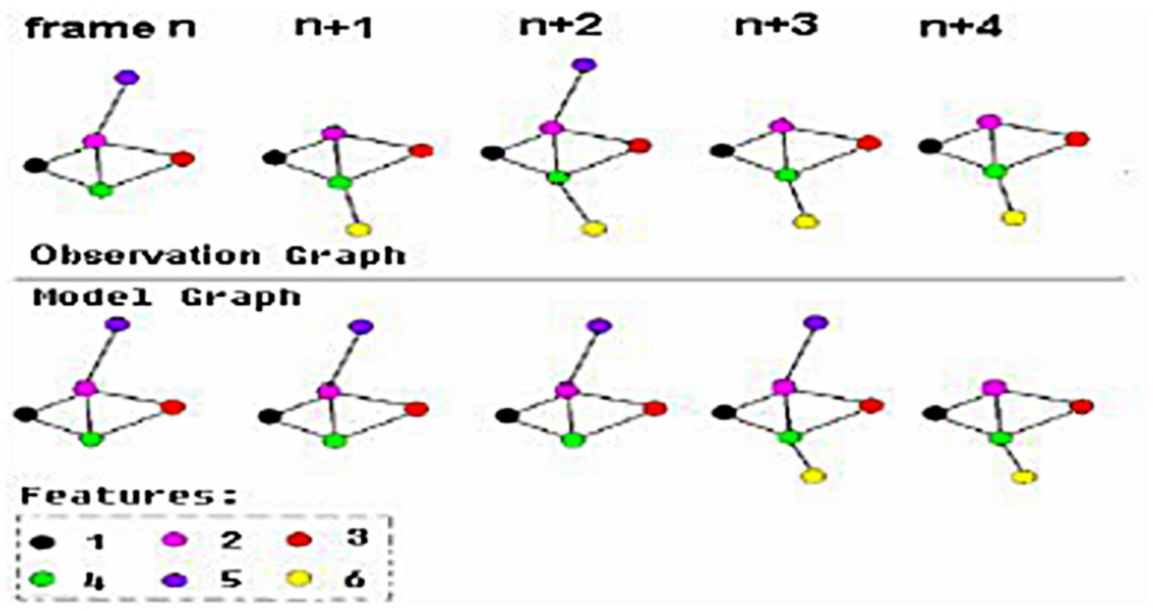


Figure 7 – A Time Lapse Demonstration Of How Dynamic Features Graph Is Updated.

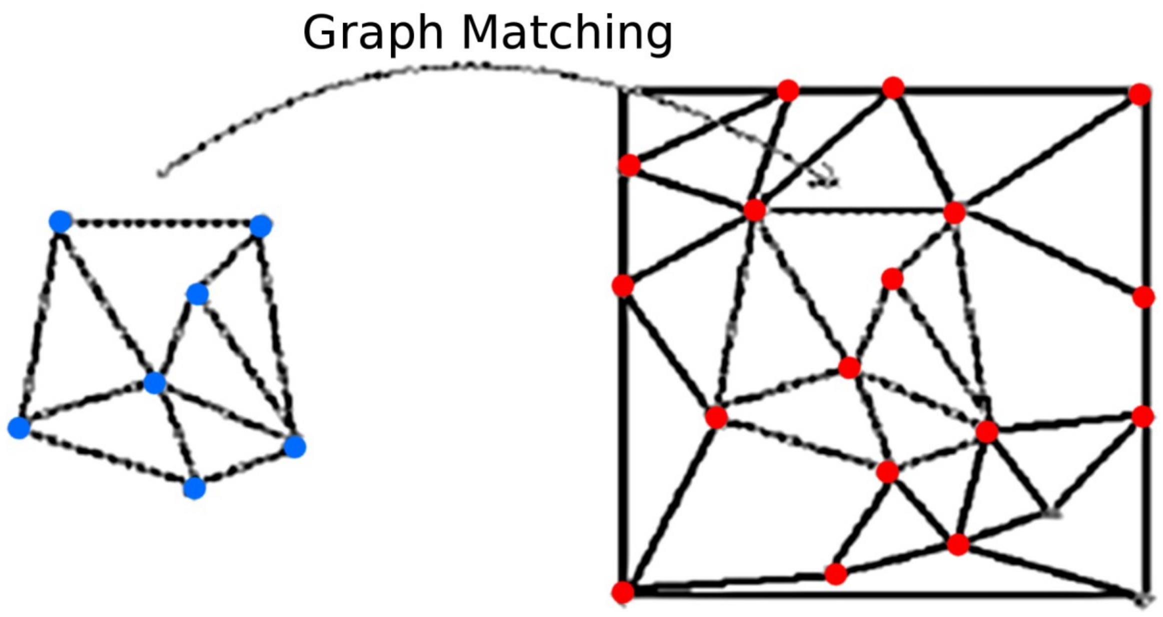


Figure 8 – Graph Matching: How Easing Helps with Matching In Feature Graph.

FLUORESCENCE

In an effort to distinguish the mussels from the bio-foul laden tank and to reliably detect them at night, we explored the idea of painting the shells. Having the shells painted has its advantages even during daytime, since there good contrast between the mussels and the sediment, there is value- added to the pictures and the time-lapse video for the scientist to be able to easily distinguish where the mussels are located and track them visually. As seen in Figure 22, painted mussel shells make for easier identification of each mussel and mussel type differentiated by color, etc.

Fluorescent paint reacts to long-wave ultra violet (UV) UV radiation, also called a black light. Paint pigments of fluorescent paint absorb the black UV radiation and radiate visible light (Harris, 2012). There are two basic kinds of fluorescent paint, visible and invisible. Invisible variation is mostly used for security purposes, like seals and marks on money, documents, etc. that are only visible when under black light. Visible paint in contrast is vibrant and visible during normal light, and glows brilliantly under black light (UVLight2).

The approach is to find a suitable visible UV pigment that would be non-toxic to the mussels and withstand being under water for prolonged periods of time and keep its fluorescent properties. Applying paint to both sides of the shell eliminates the orientation issue faced with other approaches, like printing barcodes, added weight of led packs, etc. Since the paint will cover the entire shell, some sediment buildup over time will not hinder the detection of the mussels completely in comparison to a small sensor, LED, or a printed barcode.

The experimental setup will need to be expanded to accommodate any UV light detection scheme for night time. Preferably the lights should be on before taking the pictures to charge the pigment. Once that is achieved the lights are off and then the picture is taken. This needs some high power UV lights to penetrate the amount of water in the tank and the suspended solids to adequately charge the pigment so that it can be visible in the tank.

It is preferred that the light is not left on during the time the picture is being taken. At night, the camera shutter is left on for a prolonged time to have proper exposure for development of the image. Keeping the light on during this time will flood the water with a blue tint, especially when there is high algae concentration or suspended sediment. This will effectively make detection of the mussels more challenging. Figure 22 below shows how the picture is affected when a UV light bar is turned on during night capture, there is considerable bluish tint present in on the bottom corner of the tank.

But due to bio-foul we cannot guarantee that the fluorescent dyes will get enough exposure to stay vibrant due to the excessive build up during long camera exposure at night, so we decided to keep the light on while taking the picture, this way we can reduce the exposure time and also keep the lights on for a shorter period of time to minimize the possible disruption to the mussels environment. To compensate for the blue light we will use image processing techniques to be able to detect the mussels.

Holding the light for prolonged times during the night has its adverse effects. Strong UV light will kill single cell and small micro-organisms (Hawxby). This is also the reason why it's widely used for water purification in large water tank setups and an addition for drinking water filtration systems (TROJANUV, 2012). Having the UV light on for a significant time will cause the algae in the water to be killed, this is a desired effect in show aquariums to clarify water and reduce algae buildup. It is the exact opposite effect for our experiment, since it will not be able to simulate a realistic mesohabitat.

We have also looked into possibly printing the barcodes with UV ink, doing this we would be able to detect each mussel uniquely. These approaches were explored before our initial tests in the mesohabitat. The idea would be to devise a simple tagging system that could be decoded using image processing to identify each mussel. The barcodes were printed using UV ink and stuck on the mussel shell using an adhesive. Figure 9 below shows an example of tags that were printed.

This method works well in controlled environment, but fails when the mussel orientation is concerned, so there needs to be a tag on each side. Even if we were to use two tags per mussel, there would be no solution for cases when the mussel was digs half its body in the sediment. Barcodes also need higher resolution, so when there is some bio-foul buildup, there is a high chance that the correct mussel will not be detected by image processing. So given all of these and the fact that the barcodes still had to be waterproofed after printing, did not make this a viable solution to consider in this set up. But this approach would prove itself to be a simple and elegant visual solution for more controlled environments.

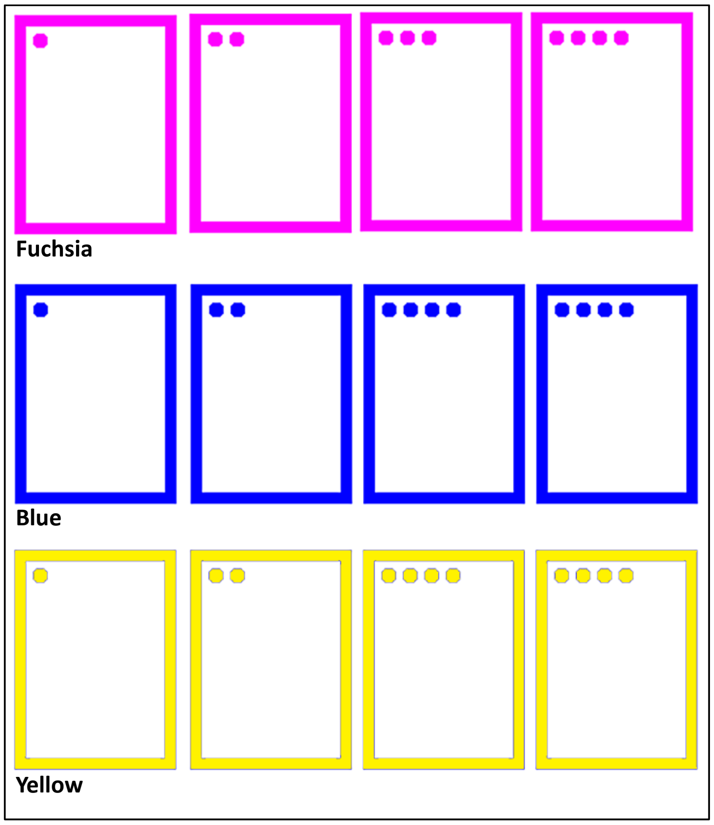


Figure 9 – Set Of UV Barcodes That Were Printed For Unique Identification Purposes.

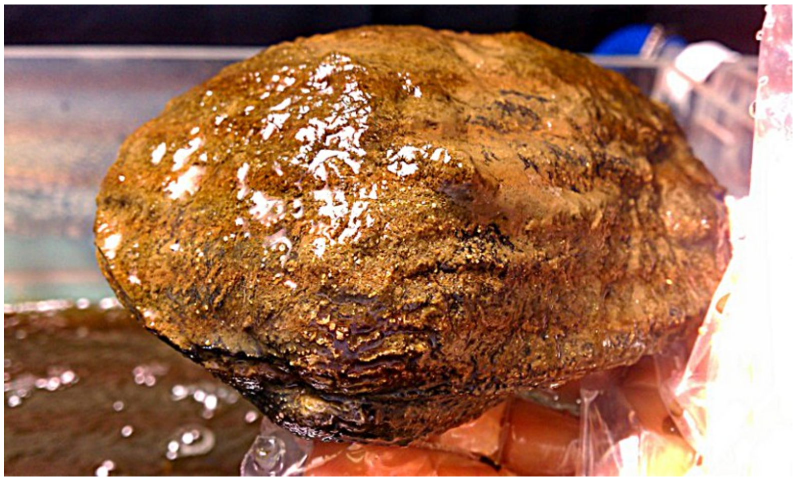


Figure 10 – Bio-foul Buildup On A Mussel: Shows The Amount Of Buildup Within A Few Days In The Tanks.

LED IDENTIFICATION

During feed cycles of the mussel tank, there is a high concentration of algae in the water, also known as green water. This causes the water visibility to diminish drastically, especially as the water gets deeper, the birds-eye view of the mussel tank does not provide enough clarity to see the sediment layer of the tank, making detection of mussels impossible without external means. However, lowering the water level by about 70% drastically improves detection, but this impacts the mussels' feeding. Figure 12 shows LED's lights penetrating through the water when there is a high concentration of algae and bio-foul suspended in the water.

Since the experiments will run for prolonged periods that will contain many cycles of feeding and possible algae blooms in the tanks, there is a need for location detection of mussels during these cycles. We explored the possibilities of using small LEDs attached to the mussels to aid visual detection. LEDs have advantages and disadvantages such as (list is not exhaustive):

- Efficiency: LEDs emit more light per watt than incandescent light bulbs. Their efficiency is not affected by shape and size.
- Color: LEDs can emit light of an intended color without using any color filters as traditional lighting methods need.
- Size: LEDs can be very small (smaller than 2 mm^2) and are easily populated onto printed circuit boards.(Kingbright, 2011)
- On/Off time: LEDs light up very quickly. A typical red indicator LED will achieve full brightness in under a microsecond. LEDs used in communications devices can have even faster response times.(Avago Technologies, 2008)
- Cycling: LEDs are ideal for uses subject to frequent on-off cycling.
- Cool light: In contrast to most light sources, LEDs radiate very little heat Wasted energy is dispersed as heat through the base of the LED.(LED6)

- Lifetime: LEDs can have a relatively long useful life. One report estimates 35,000 to 50,000 hours of useful life, though time to complete failure may be longer.(US Department of Energy, 2006)
- Shock resistance: LEDs, being solid state components, are difficult to damage with external shock, unlike fluorescent and incandescent bulbs, which are fragile.
- Focus: The solid package of the LED can be designed to focus its light. Incandescent and fluorescent sources often require an external reflector to collect light and direct it in a usable manner.

Disadvantages are:

- Temperature dependence: LED performance largely depends on the ambient temperature of the operating environment. Over-driving an LED in high ambient temperatures may result in overheating the LED package, eventually leading to device failure. Adequate heat sinking is needed to maintain long life. This is especially important in automotive, medical, and military uses where devices must operate over a wide range of temperatures, and need low failure rates.(Effects of overdrive on 5 mm LEDs, 2006)
- Voltage sensitivity: LEDs must be supplied with the voltage above the threshold and a current below the rating. This can involve series resistors or current-regulated power supplies.
- Area light source: LEDs do not approximate a “point source” of light, but rather a lambertian distribution.
- Electrical Polarity: Unlike incandescent light bulbs, which illuminate regardless of the electrical polarity, LEDs will only light with correct electrical polarity.(Avago Technologies, 2008)
- Blue hazard: There is a concern that blue LEDs and cool-white LEDs are now capable of exceeding safe limits of the so-called blue-light hazard.(Science News, 2009)

- Droop: The efficiency of LEDs tends to decrease as one increase current.(Stevenson, 2009)

Relevance To Project

To test the idea on using LEDs, we attached a number of different-colored LEDs to the shell of a dead mussel, and placed the mussel in a tank with water that has some algae. We tested each LED with light present and also in the dark. Figure 11 and Figure 12 below illustrate what was tested and captured by the camera.

This provided us with some promising data, as shown one can see the light through murky water, but that was not the only hurdle. The logistics and application of this method would prove to be more challenging that just mounting a light on the outer shell of the mussel. First, we would need to build small circuitry to house the LED and a power source; the problem with such a simple approach is that the battery will not last too long, because we will need it to be bright throughout the experiment.

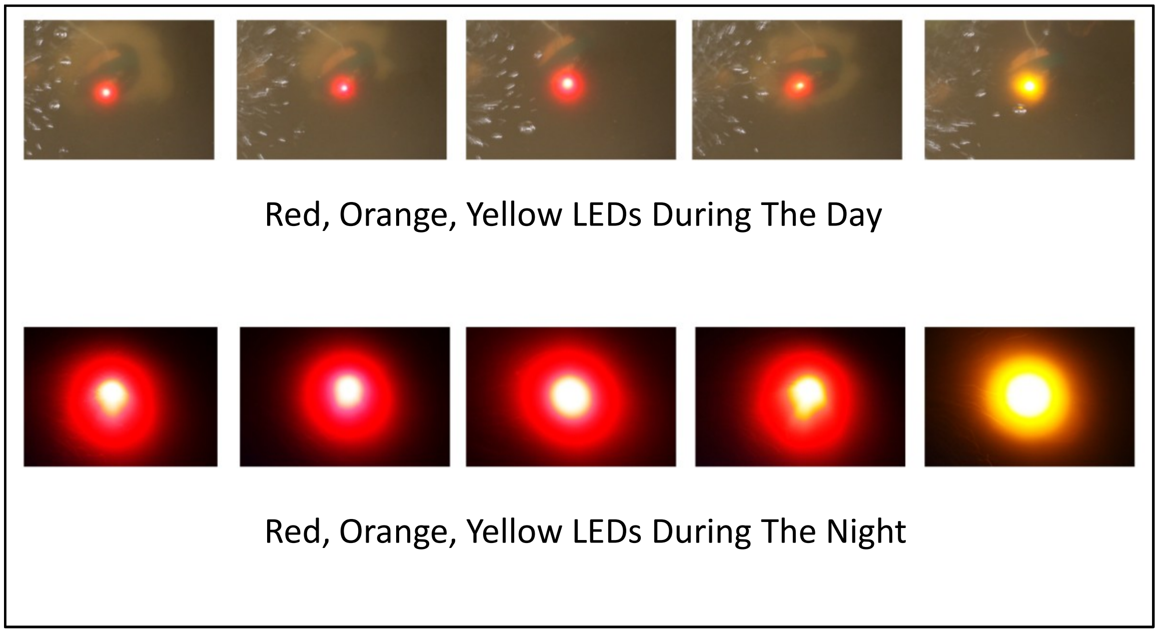

Red, Orange, Yellow LEDs During The Day
Red, Orange, Yellow LEDs During The Night

Figure 11 – Different Model LEDs Pictured At Day and Night Time.

To make the batteries last, another approach would be to turn the LEDs on at set intervals, like a timer. While this will increase battery life, there are complications. The mussel's LEDs are unsynchronized so to capture all the mussels, one would have to keep the shutter open for long periods. This is not an issue during night time, but causes overexposure of the image during daytime.

Thus, to use the LED-based tracking, one must have a method of turning on the LEDs on demand. For example, one could use a photo sensor and circuitry that will turn the LEDs on when they sense light impulse. One could then use a camera flash. Since there is a flash mounted on the camera this would not require any additions to the set up other than the light pack that will be placed on the mussels.

A hybrid approach can be to use the LEDs only at night, and during the day use colored shells as a means of identifying them. We can use different color LEDs to match the number of different coloring schemes of the shells so we can identify at least different colors, if not unique IDs. So now we have to solve the diffraction problem when the mussels are clustered closely. Larger LEDs do not have a highly concentrated beam, so the light emitted can illuminate a large area in the tank. Therefor one can have multiple mussels illuminated with several LEDs, and there is no good way to identify unique mussels, or be able to detect color accurately. Using smaller super bright LEDs with concentrated beams we can minimize this problem, but not completely eliminate.

If we were to devise very small backpacks, one can devise a simple design that turns on the LED at random delays for a short burst. This way we reduce the chance that all the LEDs will not be on at the same time, and with a short duration of light each time, diffusion and bleeding of light together can be minimized.

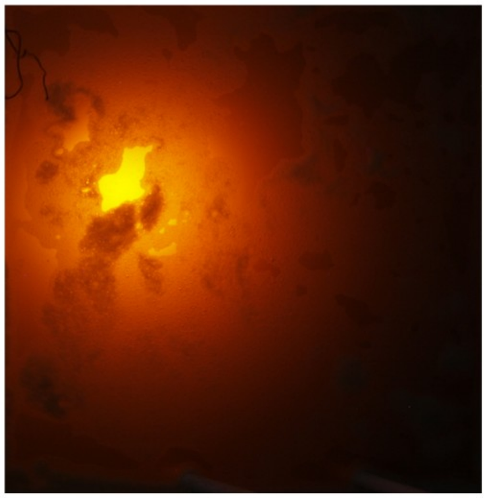
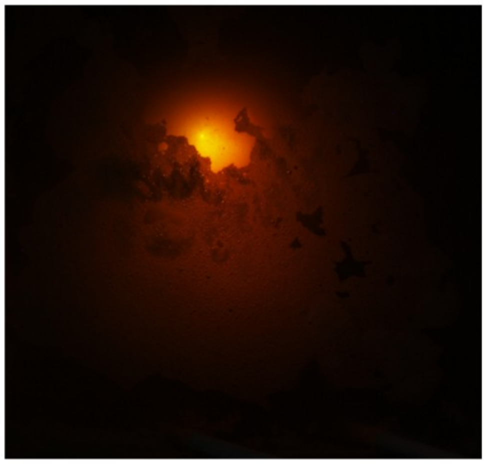


Image capturing 3 LEDs clustered together with 2 red and 1 orange LED.

With murky water conditions and longer exposure times at night cause the all the colors to merge together



Decreasing the exposure time does not help, image still looks like a single light source

Figure 12 – Picture of Clustered Mussels With Different Color LEDs with Extended and Short Exposure Times.

One way of doing this is taking multiple pictures and trying to determine which one had the expected number LEDs on at the time. This approach is very complex because it would require excessive image processing to determine which one of the images is the correct pick, the best approach would be to adjust the exposure of the camera to allow for enough time for all the LEDs to blink. If we know the upper limit of the random intervals that the LEDs blink at, we can calculate the maximum time needed for all the LEDs to be on at least once. The problem with this approach is that as the number of mussels increase, the exposure time will need to increase accordingly. This can lead to overexposure as noted earlier.

Conclusions and Remarks for LEDs

We have explored and described above some plausible approaches for identification and tracking using LEDs. Still there are additional issues. For example, highly clustered together, mussels can change orientation, or burry themselves in the sediment, that would mean that we would need to possibly place multiple LEDs on both sides of the shell.

Even if we were to solve the battery life and overexposure problems as outlined above, one major problem is the bio-foul build up that can accumulate very quickly. Figure 10 shows a picture of a mussel that has been in the mesohabitat for little over a week. There is a thick layer of buildup on the shells; even though the water might not be as murky the suspended particles will settle at the bottom continuously. This may be the biggest challenge for an autonomous system. If we even had LEDs that can have enough luminosity to pierce though a weeks' worth of bio-foul, it still needs to penetrate the cloudy water.

DETECTION THROUGH RFID

Technology Overview

Although the history of Radio frequency identification (RFID) technology can be traced to the 1930's, its roots can be traced to the invention of the radio. If one was to draw parallel to both technologies, the transponder would be the radio station, the reader would be the radio, and the antenna would determine the range of transmission. RFID technology is used to automatically identify both living and inanimate objects. (Bhuptani & Moradpour, 2005) (Lahiri, RFID Sourcebook, 2006)

Radio waves can be affected by the material through which they propagate. RF-lucent materials allow radio waves at a given frequency pass through it without any substantial loss of energy. A material is labeled RF-opaque if it blocks, reflects, and scatters RF waves. RF-absorbent materials allow the radio waves to propagate thought it but with considerable loss of energy. The RF-absorbent or opaque property of a material is relative, because it depends on frequency of the wave. That is, a material that has RF-absorbent at a given frequency can act as RF-lucent at a different range. (Lahiri, RFID Sourcebook, 2006) (Bhuptani & Moradpour, 2005)

Main classifications of RFID frequency types are listed and described below:

- Low Frequency (LF)
- High Frequency (HF)
- Ultra High Frequency (UHF)
- Microwave frequency

Low Frequency: Frequencies between 30 kHz and 300 kHz are considered low, most RFIDs use 125 kHz frequency and 134.2 kHz and higher tags and readers are uncommon. A typical LF RFID system operates at 125 kHz and generally operates using passive tags. Lower frequency ranges inheritably have low data transfer rates between the tags to the reader. LF systems are well-suited for operating environments that contains metals, liquids, dirt, snow, or mud (a very important characteristic of LF systems). Due to this property, LF systems are used widely for animal identification, access control and industrial automation. Because of the maturity of these types of tags and the range is accepted worldwide, LF tag systems probably have the largest installed base. (Lahiri, RFID Sourcebook, 2006)

High Frequency HF frequencies range from 3 MHz to 30 MHz, with 13.56 MHz being typical frequency used for HF RFID systems. High frequency systems do not perform well near metals, but have relatively faster transfer rater and longer range, a maximum of 1 m or more. HF systems are also widely used, especially in hospitals, payment and royalty cards, smart shelf, etc. (Bhuptani & Moradpour, 2005)

Ultra High Frequency: UHF ranges in between 300 MHz to 1 GHz. Typical passive UHF RFID system in the United States operates at 915MHz, and a typical active UHF RFID system operates at 315 MHz and 433 MHz. Active UHF offers very long transmit ranges, up to 100 meters, and passive tags can be read from over 3 meters. UHF systems also have high data transfer rates; coupled with its distance capabilities they are well suited for supply chain management and logistics. (Lahiri, RFID Sourcebook, 2006)

Microwave Frequency: Microwave frequency ranges upwards from 1 GHz. A typical microwave RFID system operates either at 2.45 GHz or 5.8 GHz, with later being more common. Microwave range can use both semi-active and passive tags, has the fastest data-transfer rate between the tags and read ranges all similar to UHF systems. Usually used in electronic toll collection. Microwave along with UHF perform poorly around metals and liquids. (Lahiri, RFID Sourcebook, 2006)

RFID System

An RFID system can be described as an integrated collection of components that implement an RFID solution. Components listed below outline the most common components of an RFID system (Lahiri, RFID Sourcebook, 2006):

- Tag
- Reader
- Reader Antenna
- Controller
- Sensor, actuator
- Host and software system
- Communication infrastructure

Communication infrastructure is the backbone of the whole system; it facilitates data transfer from one module to the next via multiple communication protocols and interfaces like RS-232, Ethernet, etc.

The Figure 13 below shows the diagram of an RFID system with all its components.

Tag

An RFID tag, also known as a transponder is a device that can store and transmit data to a reader in a contactless manner using radio waves when interrogated by the reader. Tags can be classified into two main categories, without on-board power and with. The types of tags and their descriptions are listed below: (Lahiri, RFID Sourcebook, 2006)

- Passive
- Active
- Semi-active

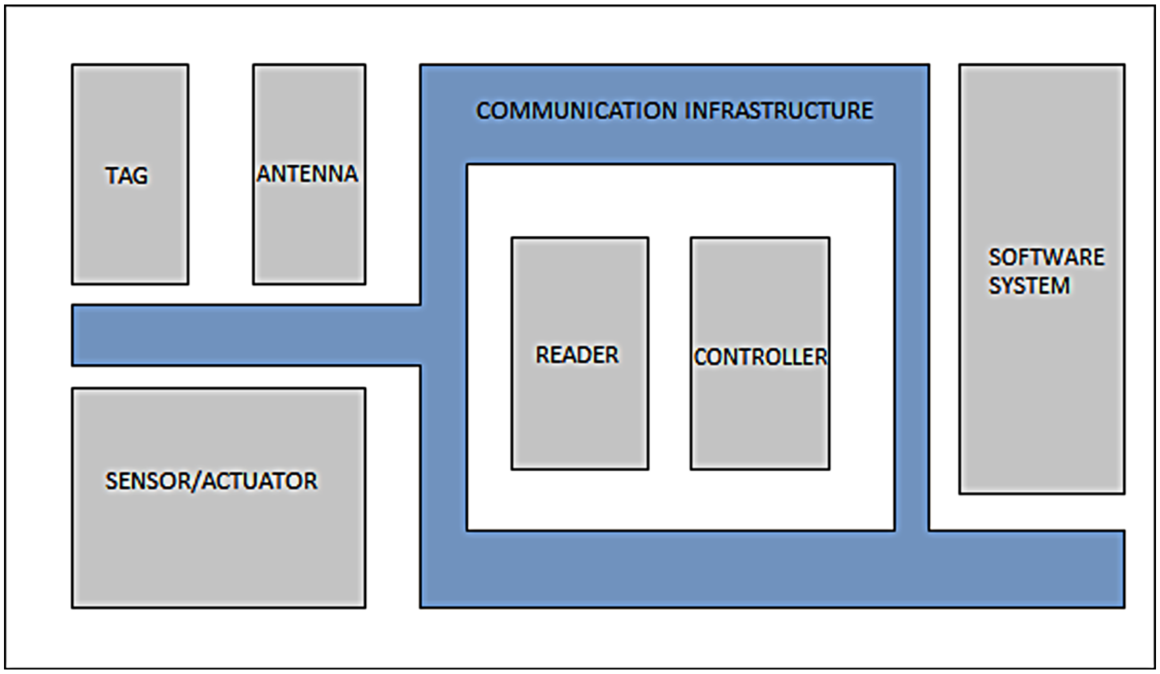


Figure 13 – Schematic View of RFID System That Outlines All Of The Main Components. (Lahiri, RFID Sourcebook, 2006)

Passive Tags: This type of RFID tag is the most popular tag being used today, and does not have an onboard power source. These tags communicate via harnessing the power emitted from the reader to energize itself and transmit its stored data. A passive tag has a simple construction, comes in a single package and as a result, such a tag has a long life and is generally resistant to tough environmental conditions. (Lahiri, RFID Sourcebook, 2006)

To communicate with a passive tag the reader always communicates first, followed by the tag. The presence of a reader is mandatory for such tag to transmit its data. It has a variety of read ranges starting with less than 2.5 cm to about 10 m. A passive tag consists of the following main components: Microchip and Antenna. Passive tags are typically smaller than active or semi-active tags, which are dependent on on-board power source. (Bhuptani & Moradpour, 2005) (Lahiri, RFID Sourcebook, 2006)

Figure 14 below shows the basic components of a microchip. The power control/rectifier converts AC power from the reader antenna signal to DC power. It supplies power to the other components of the microchip. The clock extractor extracts the clock signal from reader antenna signal. The modulator modulates the received reader signal. The tag's response is embedded in the modulated signal, which is then transmitted back to the reader. The logic unit is responsible for implementing the communication protocol between the tag and the reader. The microchip memory is used for storing data. Continuous advances in technology continue to shrink the size of the microchip. One must note that the tag's physical dimensions are not determined by the size of its microchip but rather by the size of its antenna which will determine the read range. (Lahiri, RFID Sourcebook, 2006)

Active and Semi-Active Tags: Active RFID tags have an on-board power source. An active tag uses its on-board power supply to transmit data to a reader. It does not need the reader's emitted power for data transmission. Due to the presence of power, these tags are usually coupled with other electronics for performing certain tasks. The on-board electronics can contain microprocessors, sensors, and input/output ports powered by the on-board power source. One way to think of an active tag is a wireless task specific computer. (Lahiri, RFID Sourcebook, 2006)

A semi-active tags are very similar to active tags that they are suited for performing specialized tasks. The on-board power supply provides energy to the tag for its operation. However, for transmitting its data, a semi-active tag uses the reader's emitted power. Using modulated backscatter scheme the semi-active tag can have considerable range of operation, about 30 meters under ideal conditions. We will not go into details of the technology used for these tags since it is not utilized in our project. (Lahiri, RFID Sourcebook, 2006)

Tag Antenna: A tag's antenna is used for harnessing energy from the reader's signal to energize the tag and for sending and receiving data from the reader. This antenna is physically attached to the microchip and this geometry is central to its operation. Infinite variations of antenna design are possible, and designing an effective antenna for a tag is considered as much as an art as a science. The antenna length is directly proportional to the tag's operating wavelength, the construction of an antenna will affect it sensitivity of orientation, and range. (Lahiri, RFID Sourcebook, 2006)

Readers

An RFID reader is also called an *interrogator*, and is a device that has the capability to process data received from the tag. The reader also provides for a communication interface with the host computer. Although the reader can also write to a tag, it is still called a reader. The reader has four main functions. These are (a), energize a tag by providing the power to the reader antenna, (b) define the operating frequency, (c) read data from the tag, and possibly write data to the tag and communicate with the host computer. Writing the tag data by a reader is called *creating a tag*. The process of creating a tag and uniquely associating it with an object is called *commissioning the tag*. The reader is the central nervous system of the entire RFID hardware system. (Lahiri, RFID Sourcebook, 2006)

Transmitter: The transmitter of the reader is designed to transmit AC power and clock cycles at the selected frequency via its antennas to the tags in its read zone. This part of the transceiver component is responsible for sending the reader's signal to the surrounding environment and receiving tag responses back via the reader antenna(s). The antenna ports of a reader are connected to its transceiver component. One reader antenna can be attached to each such antenna port, but there are advanced readers that can have multiple ports available. (Lahiri, RFID Sourcebook, 2006)

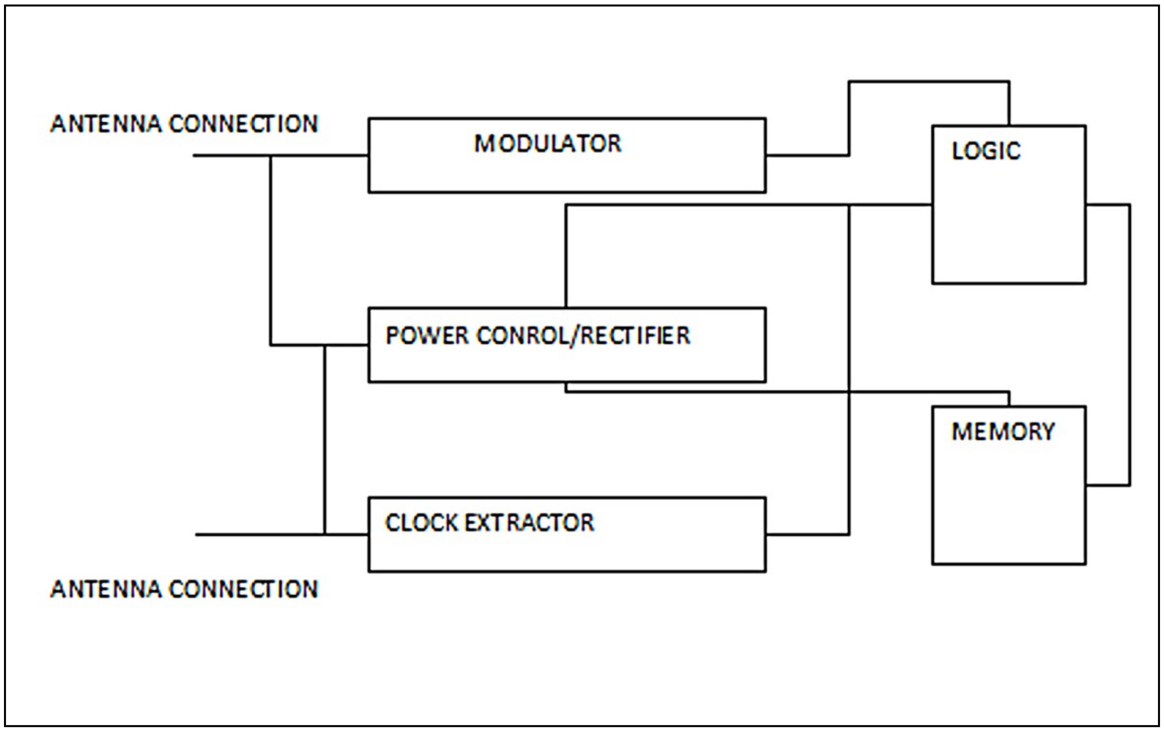


Figure 14 – Block Diagram Showing RFID Microchip Components. (Lahiri, RFID Sourcebook, 2006)

Receiver: This component is also part of the transceiver module. It receives analog signals from the tag via the reader antenna. It then sends these singles to the reader microprocessor, where it is converted to its equivalent digital form, that is, the digital representation of the data that the tag has transmitted to the reader antenna. (Lahiri, RFID Sourcebook, 2006)

Microprocessor: This component is responsible for implementing the reader protocol to communicate with compatible tags. It converts the analog signals to its digital representation and performs a number of decoding and error checking algorithms of the received analog signals such as collision detection etc. In addition, more advanced microprocessors can contain custom logic for doing low-level filtering and processing of the incoming data. (Lahiri, RFID Sourcebook, 2006)

Controller: A controller is a module that allows an external entity, human or computer, to communicate with and *control* a reader's functions and to control existing actuators, if any, associated with this reader. Often, manufacturers integrate this component into the reader itself. However, it is also possible to package this as a separate hardware/software component that must be bought together with the reader. (Lahiri, RFID Sourcebook, 2006)

Communication Interface: The communication interface component provides the communication instructions to a reader that allows it to interact with modules external to it, by the logic in the controller it transfer its stored data and accepts commands and sends back the corresponding responses by a defined protocol. Some might see this as a functionality of the controller but this entity has important characteristics that make it necessary to treat this as an independent component. A reader could have a serial as well as a network interface for communication. A serial interface is probably the most widespread type of reader interface available, and this is what we are using in our project. (Lahiri, RFID Sourcebook, 2006)

Serial Reader: Serial readers use a serial communication link to communicate with an application. The reader is physically connected to a computer's serial port using an RS-232 or RS-485 serial connection. Both these connections have an upper limit on the cable length that can be used to connect a reader to a computer. The advantage of serial readers is that the communication link is reliable compared to network readers. Therefore, the use of these readers is recommended to minimize dependency on a communication channel. The disadvantage of serial readers is the dependence on the maximum length of the cable that can be used to connect a reader to a computer. The serial data-transfer rates are slower for serial readers compared to network readers and the limitation of serial ports available on host machines is also a factor. (Lahiri, RFID Sourcebook, 2006)

Communication between a Reader and Tag

Depending on the tag type, the communication between a reader and a tag can be one of the following: modulated backscatter, transmitter type, transponder type. The type of tag and antenna type to use is determined by the concepts of near and far field. Simply put, the area between a reader antenna and one full wavelength of the RF wave emitted by the antenna is called the near field. The area beyond one full wavelength of the RF wave emitted from an antenna is called the far field. Passive RFID systems operating in LF and HF use near field communication, whereas those in UHF and microwave frequencies use far field communication. The signal strength in near field communication attenuates as the cube of the distance from the reader antenna. In far field, it attenuates as square of the distance from the reader antenna. As a result, far field communication is associated with a longer read range compared with near field communication. (Lahiri, RFID Sourcebook, 2006)

Since we are utilizing LF frequency range of operation, we will not go into detail of transmitter and transponder type communications, but will expand on modulated backscatter communication.

Modulated Backscatter

Modulated backscatter communication is the type of communication where the reader sends out a continuous wave RF signal containing AC power and clock signals to the tag at the carrier frequency. This type of communication applies to passive as well as semi-active tags. Power is supplied to the tag's microchip through physical coupling. About 1.2 V is generally necessary to energize the tag microchip for reading purposes. The microchip now modulates or breaks up the input signal into a sequence of on and off patterns that represents its data, and transmits it back. When the reader receives this modulates signal, it decodes the pattern and obtains the tag data. Thus, in modulates backscatter communication, the reader always initiates communication first, followed by the tag. A tag using this scheme cannot communicate at all in the absence of a reader because it depends totally on the reader's power to transmit its data. (Lahiri, RFID Sourcebook, 2006) (Lahiri, RFID Sourcebook, 2005)

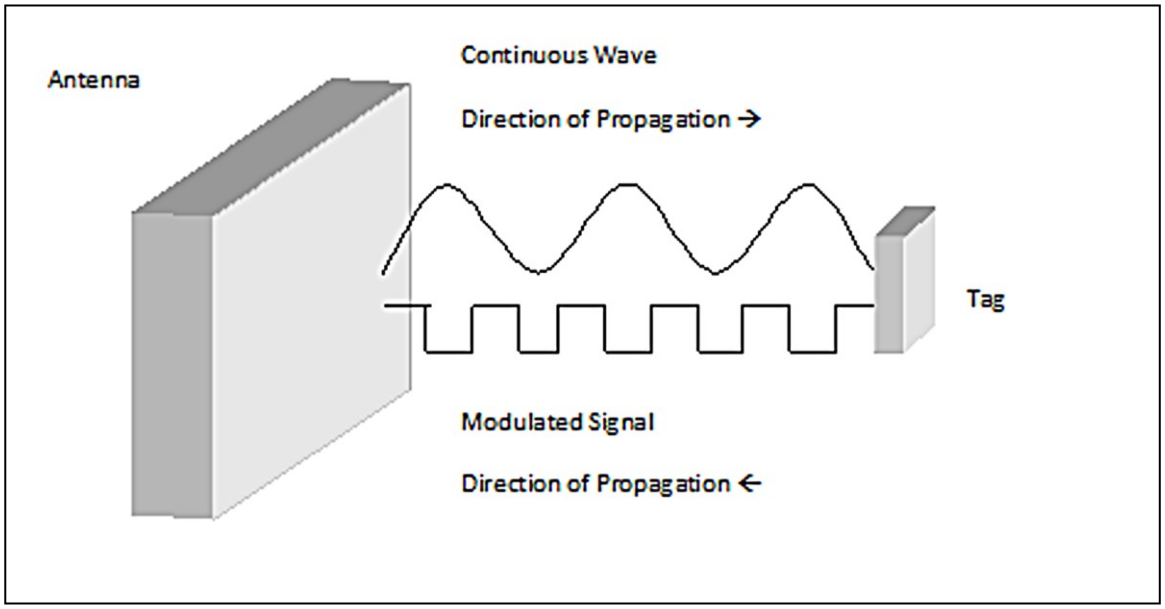


Figure 15 – A Visual Demonstration of Modulated Backscatter (Lahiri, RFID Sourcebook, 2006)

Reader Antenna

A reader communicates with a tag through the reader's antennas, a separate device that is physically attached to a reader, at one of its antenna ports. Because the antenna creates an electromagnetic field to couple with the tag, a reader antenna is also referred to as the reader's *coupling element*. An antenna broadcasts the reader transmitter's RF signal into its surroundings and receives tag responses on the reader's behalf. Therefore, proper positioning of the antennas is essential for good read accuracy.

Antenna Footprint: The footprint of an antenna is a three-dimensional region shaped somewhat like an ellipsoid or a balloon projecting out of the front of the antenna.

The footprint of the reader's antennas, also referred to as the *antenna pattern*, determines the read zone of a reader. In this defined region, the antenna's energy is most effective; therefore, a reader can read a tag placed inside this region with the least difficulty. In reality, because of antenna characteristics, the footprint of an antenna is never uniformly shaped but always contains deformities or protrusions. Each protrusion is surrounded by dead zones. Such dead zones are also called nulls. Figure 16 below show a simple antenna pattern and a pattern with protrusions. (Lahiri, RFID Sourcebook, 2006) (Bhuptani & Moradpour, 2005)

A tag placed in one of the protruded regions will read, but if this tag moves slightly so that it is inside the surrounding dead region, the tag cannot be read, which leads to counter-intuitive reading behavior. For example, when placed a certain distance away from a reader; a tag does not read, but when moved slightly in one direction it can be picked up by the reader; if this tag is moved even slightly in another direction, it may not be detected again. Therefore, when an antenna is selected to cover a read area, it is important that one does not depend of the protruded regions to maximize the read distance.

Antenna Polarization: As the antenna emits electromagnetic waves into its surroundings, the direction of oscillation of these electromagnetic waves is called the polarization of the antenna. The polarization of the antenna has major effects on the readability of the tags. Based on polarization there are two main types of antennas Linear and Circular polarized.

In linear antennas, RF waves emanate in a linear pattern from the antenna. These waves have only one energy field. A linear polarized antenna has a narrower radiation beam with a longer read range. In addition, a narrower radiation bean helps a linear polarized antenna to read tags, within a longer, narrow but well-defined read region. However, a linear polarized antenna is sensitive to tag orientation with respect to its surroundings. (Lahiri, RFID Sourcebook, 2006)

RF waves radiate from a circular polarized antenna in a circular pattern. These waves have two constituting energy fields that are equal in amplitude and magnitude, but have phase difference of 90 degrees. Therefore, when a wave of an energy field is at its highest value, the wave of the other field is at its lowest. Figure 16 below shows the resulting pattern from a circular polarized antenna. Because of the nature of polarization, a circular polarized antenna is largely unaffected by tag orientation. Therefore, this type of antenna proves ideal for applications where the tag orientation is unpredictable. (Bhuptani & Moradpour, 2005) (Lahiri, RFID Sourcebook, 2006)

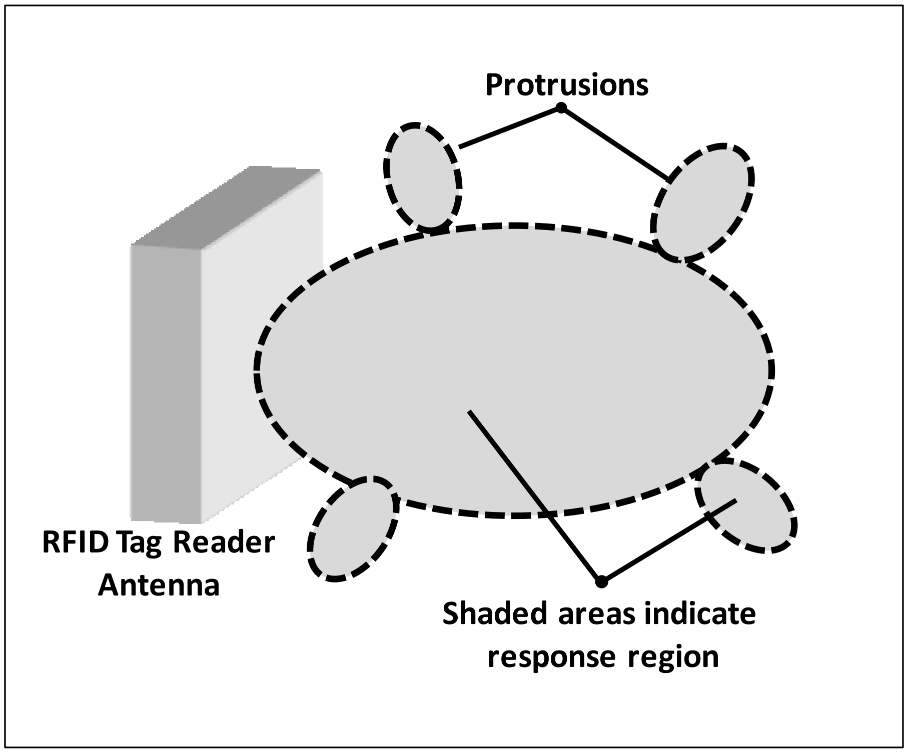


Figure 16 – Diagram Showing Antenna Protrusions And Dead Zones Inherit in Antennas. (Lahiri, RFID Sourcebook, 2006)

Integration and Consideration of RFID to Project

Our goal was to use RFID tags to uniquely identify each of the mussels and see how feasible it was to detect their location on the *x*/*y* plane. Image processing alone was not going to suffice in long term experiments due to factors mentioned in earlier chapters like bio-foul buildup, murky water, etc. RFID technology is mature and there is a large variety of tags and readers available. We needed to find what kind of solution would work our experimental environment. We needed to penetrate plywood, thick vinyl fish tank walls, a few centimeters of wet sediment, and depending on the orientation of the mussel, approximately another 2–3 cm of water.

We found it difficult to acquire certain RFID, tags because the vendors would only supply to customers with larger orders. Some of the more capable tags and readers (e.g., capable of handling tag collision, longer range), are only available in orders of 500 or more..

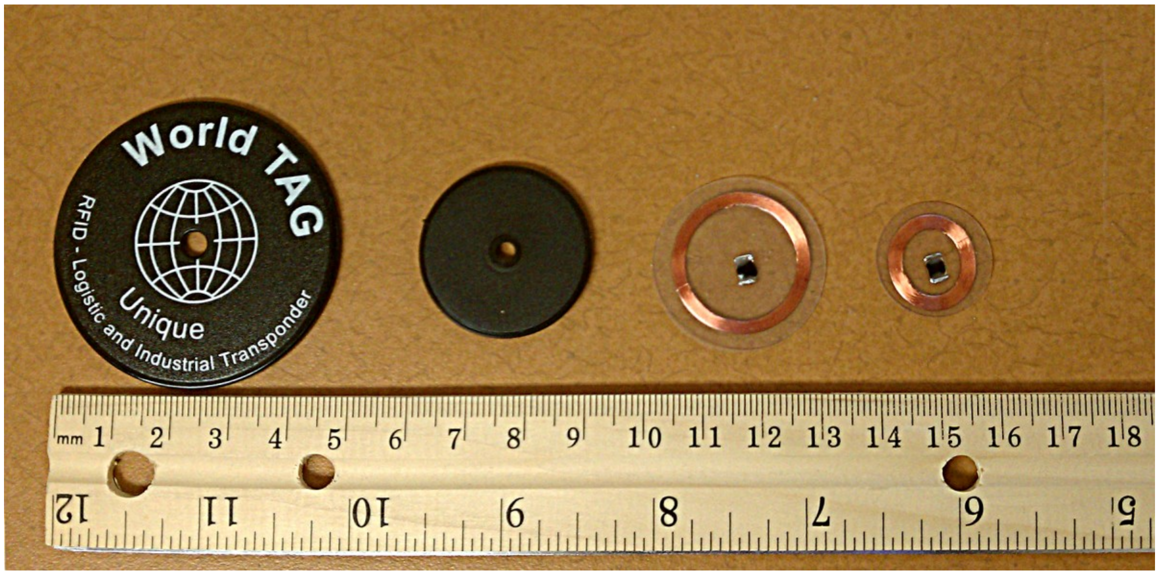


Figure 17 – Different Size RFID Tags Experimented With During Testing Phase.

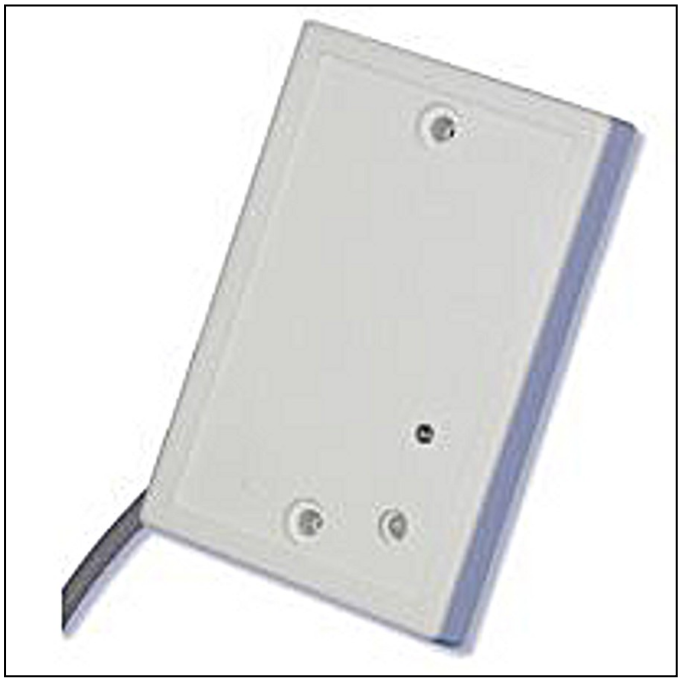


Figure 18 – The RFID Reader Used In Our Project, It's Mounted On The *x*/*y* Traverse.

In the earlier phases of testing we used RFID kits from small suppliers. . One particulate kit operated at 125 kHz, and s worked well outside of the water and we were able to get at least 6 cm. The size of the antenna on the reader made to make a significant difference. For example, the smallest tag we had is typically used in live animal tags. Given their intended purpose, they were very small. This seemed like an ideal tag for us, due to its size and weight. However, we found that once in the water, the tags could not be read. Even with no sediment, we could not penetrate the walls of the fish tank and the plywood base. Contrast a keycard tag, the size of a credit card, was not really considered for this experiment, but it was interesting to see if it would work, having a larger surface area due to a larger antenna. It was able to penetrate the base, walls of the fish tank and about 8 cm of sediment.

The major shortcoming of our initial test set up was the inadequacy of the readers we were using; they were very low power and had small antennas. So we needed a new approach for the antenna as well we were searching for other tags as well. There are not many low-frequency-readers available in the market. Most applications are for hobby kits, hand-held, or identification for security. The security identification readers were more readily available, and provided more flexibility because we could get units that were through-wall. Vendors provided models that worked with higher power antennas, so that there was no need for physical contact, etc.

Medium	**Tag**	**Flat**	**Diagonal**
AIR	5 cm Through Hole	~ 26.5 cm	~ 11.5 cm
AIR	3 cm Through Hole	~ 19 cm	~ 10.2 cm
AIR	3 cm Clear	~ 14.6 cm	~ 8.3 cm
AIR	2 cm Clear	~ 10.8 cm	~ 6.4 cm

Table 1 – Table Capturing The RFID Read Ranges As Air For The Medium.

We ordered some industrial circular tags, some real thin and clear and the others with industrial plastic coating with through holes. Both waterproof and with larger antenna footprints as shown in Figure 17 above. When coupled with high power antennas our distance measurements in air and water increased significantly. The Table 1 above shows the measurements taken when the tags were held parallel and diagonal to the reader. The following image shows the test setup the measurements were taken in. This gave us the confidence that the RFID solution is going to be a plausible approach. We also needed to overcome known shortcomings of this technology such as tag collision, how one would find corresponding *x* and *y* coordinates of each tag, and what kind of spatial resolution we would achieve.

Referring to Figure 16, the RFID tag reader reliably reads a tag within an ellipsoid. Further, the reader will in some instances read tags that are located within the protrusions shown. However, reading tags located within these protrusions, is intermittent and not reliable. This is an inherent shortcoming of the RFID technology. In traditional applications of RFID (asset tracking, etc.) this is not a major concern. However, our aim is to use RFID to provide *location* information, and the protrusions add to the uncertainty of our location estimation.

Medium	**Tag**	**Flat**	**Diagonal**
Sediment/Water	5 cm Through Hole	~10 cm	~10 cm
Sediment/Water	3 cm Through Hole	~6.4 cm	~ 5 cm
Sediment/Water	3 cm Clear	~5 cm	~ 5 cm
Sediment/Water	2 cm Clear	~1.8 cm	~0.5cm

Table 2 – Table Capturing RFID Read Ranges In Sediment And Water As The Medium.

To get location of each mussel, the best way is to move the reader under the tanks to detect present tags. To achieve this we utilize two motor- controlled slides, one for each axis that will carry a reader as its payload. The slide’s location can be controlled and changed by the computer. The application sends commands to move both motors to change the location of the reader. Figure 20 shows the slide that resides under the mount and the following flowchart in Figure 21 shows how the application layer controls the slides. Each tank has equidistance dimensions of 60 cm, so the traverse will have to span a minimum distance of 48×24 cm, so the slide we have acquired goes a little longer than this make sure we have enough clearance to cover both tanks adequately.

An issue is that if we were to keep the power of the reader constantly, we will cause collisions and also read tags that are not necessarily under the reader due to the protrusions’ that the antenna generates. This will reduce our ability to detect location data with any credible and useful resolution. For reading tags that are energized and are not under the reader calls for more complex measures.

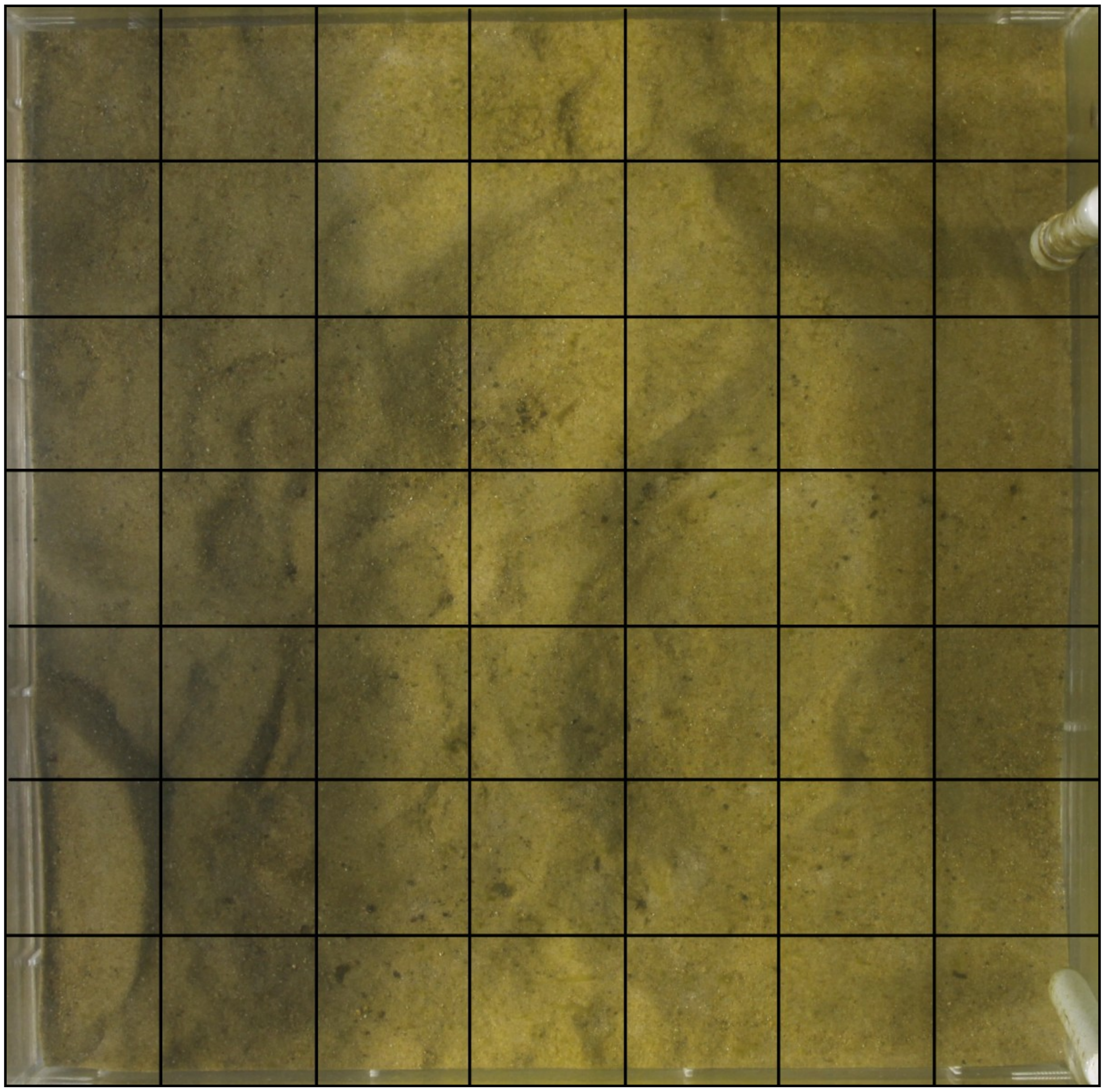


Figure 19 – Overlay of Virtual 9 cm×9 cm Segments Showing How The Traverse Will Scan The Tank For RFID Detection.

The resolution of movement will be fixed in a grid format, as shown in Figure 19. By controlling the power to the reader, we will not energize tags in transition to a new location. This also provides us to define a predefined set of locations. Power to the reader and slide motors are under computer control. The overall interface to the traverse and the RFID system is shown in the flowchart in Figure 21.

We have thus far solved an important problem, reading a tag that is under the reader rather than around it. But now we need to solve a tougher question. What happens if there are multiple tags under the reader? Is the technology capable of solving such issues? Since there are size-constraints, it is highly unlikely that we will have more than 3 mussels at a time that can be at the active area of the readers' antenna. We ran some simulated tests of such a scenario, and ran the antenna under the tank to see how many tags were detected and how many iterations it took. Interestingly enough, that there are cases when different tags are picked up when power is cycled to the antenna, we have used this to our advantage.

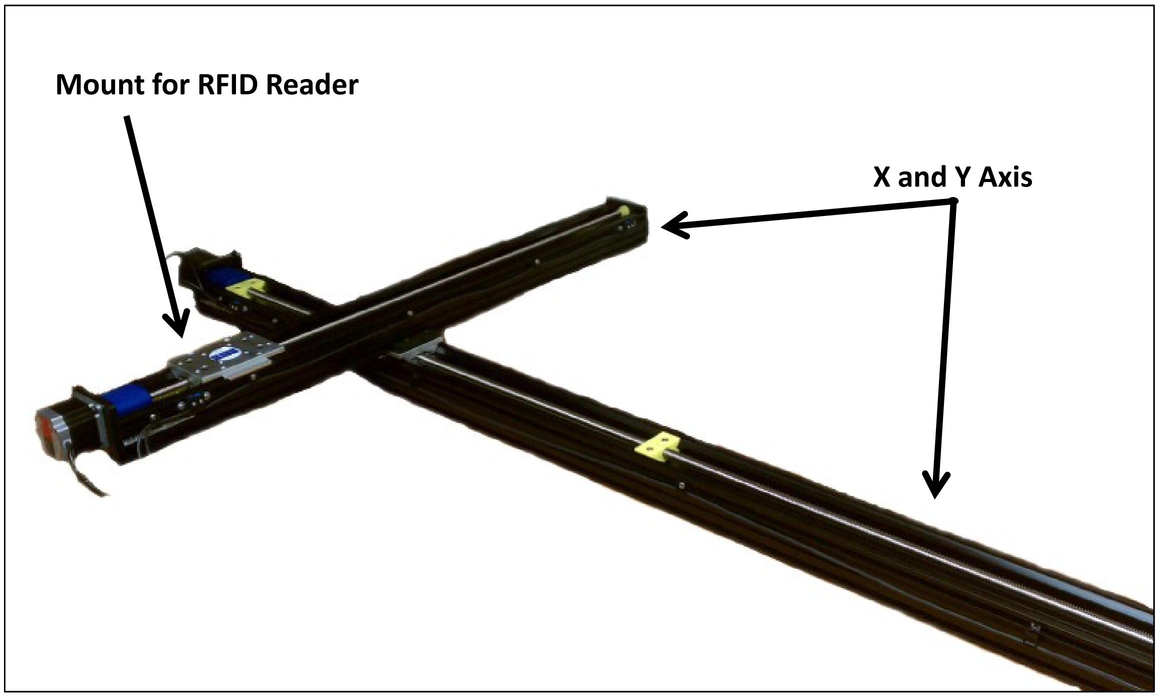


Figure 20 – The *x*/*y* Traverse Located Under The Mussel Tanks.

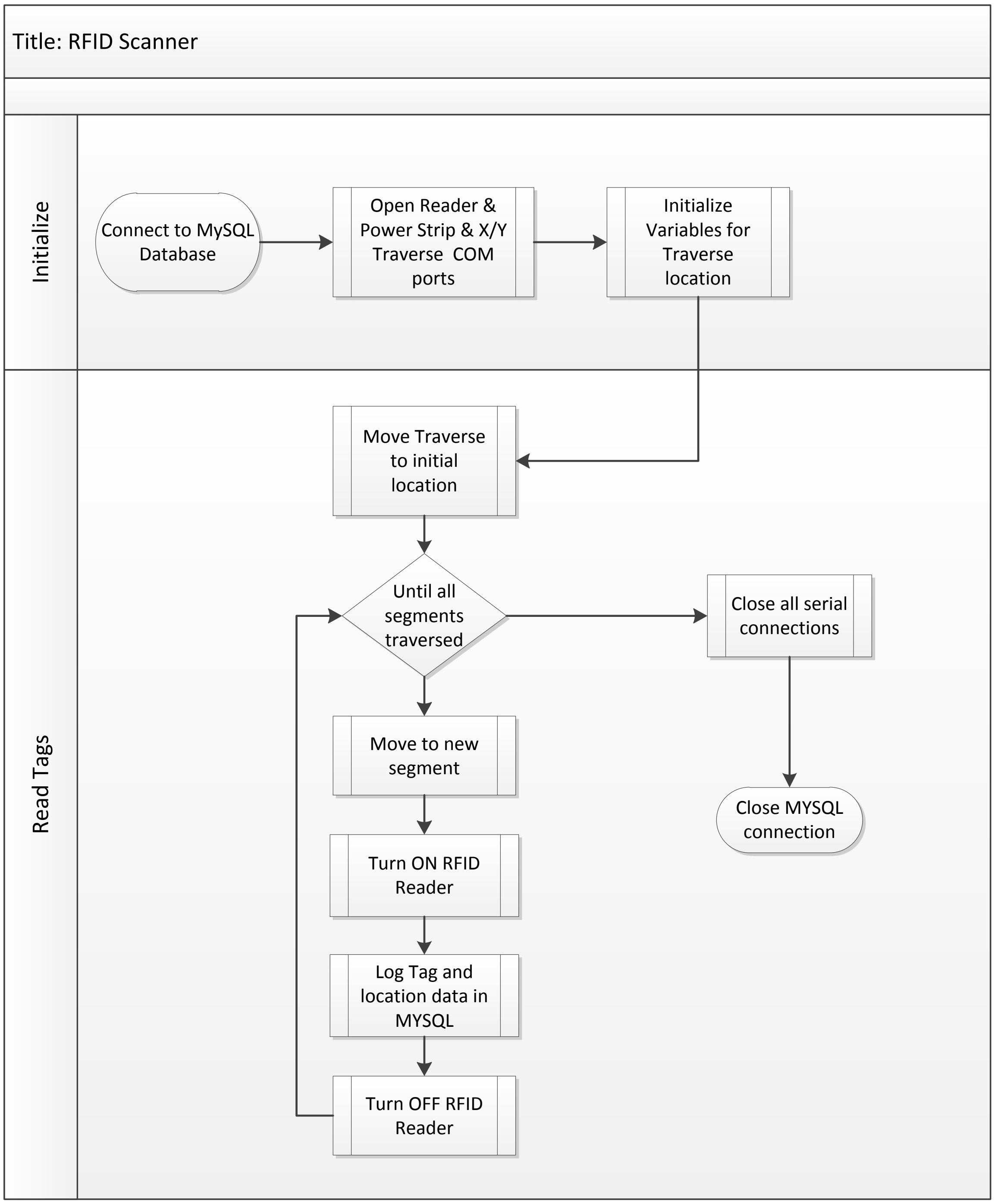

Title: RFID Scanner
Initialize
Connect to MySQL Database
Open Reader & Power Strip & X/Y Traverse COM ports
Initialize Variables for Traverse location
Read Tags
Move Traverse to initial location
Until all segments traversed
Close all serial connections
Move to new segment
Close MYSQL connection
Turn ON RFID Reader
Log Tag and location data in MYSQL
Turn OFF RFID Reader

Figure 21 – Flowchart For PYTHON Script That Control RFID Scanner.

IMAGE PROCESSING

To track mussels visually, the best approach is to use unique colors. We took a practical approach to color 3 mussels of interest with different colors and the rest of the mussels were colored the same. The idea is to see how the 3 mussels move individually, and track the rest as a group to visually to see if there is an interesting clustering behavior. Figure 23 below shows an example of how the shells were painted.

There are two tanks side by side, one with the backpacks and the next is the control tank. The camera is adjusted to capture both tanks from above. Simple image manipulation is done to isolate each tank. Since the camera and the tanks are stationary, utilizing fixed transformations to isolate the two tanks will suffice. After the initial step we have generated two images from a single one. Exactly the same image processing steps are applied to both images.

For image processing, we utilize an application called *FIJI* (FIJI, 2012), which is basically a wrapper around the *ImageJ* libraries (ImageJ, 2012). This is a very powerful toolkit and is capable of providing all the functions needed for our experiments. The sediment in the tank is non-uniform it is composed of coarse broken rock of multiple colors and different sizes, etc. Although the picture is of real high quality, the sediment simulates areas of high noise this is further affected by the varying water quality. To address this, we apply a Gaussian Blur Filter (Gonzales & Woods, Image Enhancement in the Frequency Domain, 2002) to blur out the grains and also to smooth out the heterogeneity of the mussels. The image itself is large, so we use a large sigma value of 6. A smaller value will affect the segmentation of the image due to noise patterns. Too high of a value and the image is too blurry to process any edge information out of it.

We are working with colored images, so it makes it more challenging to detect unique colors. If we were to simply detect objects regardless of color, we could have used pixel intensity threshold method on the image after extracting the brightness information or just converting the image to 8-bit. Our intention is to detect at least a few different colors reliably and trace their movement along with a large number of mussels with the same color. So for each color we create a new mask using RGB Color Threshold functions, and specifying Red, Green and Blue ranges for each color to be tracked. Images below illustrate these methods.

For color detection, we experimented with different color spaces and methods. We found that if we were to not track individual colors (even with color thresholding) knowing that the background is darker than the foreground of the highly vibrant paint, the automated method works very well. The results on the data set were very promising. All the mussels were detected every time reliably. However, this simple method will not work if we were to filter out each color. Particular ranges of Red, Green, and Blue have to be picked. We have to define 4 different types of RGB ranges. These ranges are used to create thresholds that are run on the base image for to detect each unique color.

In addition to this, the ranges have to be modified at night. We can detect all the mussels at night by utilizing a UV-lighting, similar to the ones used to detect counterfeit money. So for images that are taken at night, have a bluish tint so using the same thresholds do not provide correct results, Figure xxx below shows the outcome of a night image to detect the color orange. It’s clear to see that we need to be able to detect the two different categories of images, day and night.

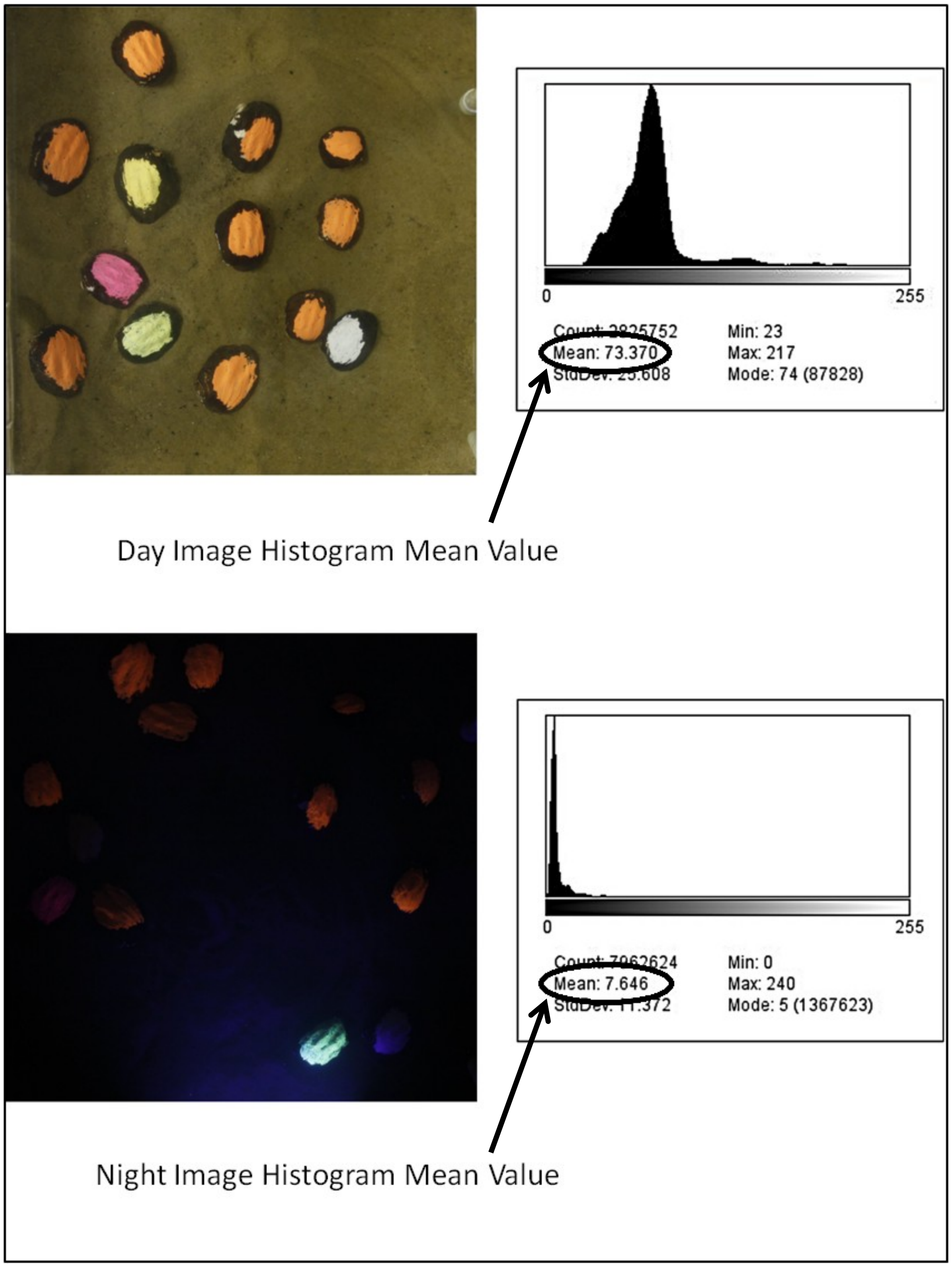


Figure 22 – Histogram of Day and Night Pictures: Its Clear To See The Difference Between Day And Night Pictures.

The day images have significant more light. This causes the camera sensor to be able to capture a lot more detail; this means much more data in general to be stored in the file. In contrast, darker photos lack detail and color. . This reflects on the histogram. A histogram of an image is the tonal distribution in an image. It is a plot of the number of pixels of each tonal value (Gonzales & Woods, Image Enhancement in the Spatial Domain, 2002). The horizontal axis of the graph represents the tonal range and the vertical axis represents the number of pixels in that that particular tone. Figure 22 below shows images with its corresponding histograms.

The day images have a wide range of tone, and one can observe a wide range of distribution on the tonal axis between 0 and 255. The night images are closer to 0. Consequently, it is very easy to determine if the picture being processed was taken at day time or at night by evaluating the mean value. The mean of the day pictures are always above 50, we know what any picture that has a mean value smaller than that has been taken at night with UV light, so we can apply our night threshold ranges to these images.

Day threshold values are easier pick because the histogram has wider range of tones for each color and there are enough non-overlapping tones so we can select blobs relatively clean. Figure 25 shows a day image's histograms of red, green and blue. . Filtering out the white shells is one of the more challenging colors to detect in the tank, especially at night because it takes on the bluish tint of the UV light. The white in the day time is easy to detect, and the threshold image is relatively clean. In contrast, Figure 25 also shows the night red, white and blue histograms. The data is highly skewed to the dark end of the scale and very narrow, this cause data that is of no interest, such as the outer circumference of other shells and other "noise". The figure shows a threshold of a night image and the white threshold. It is clear that many unwanted pixels make it through the filter.

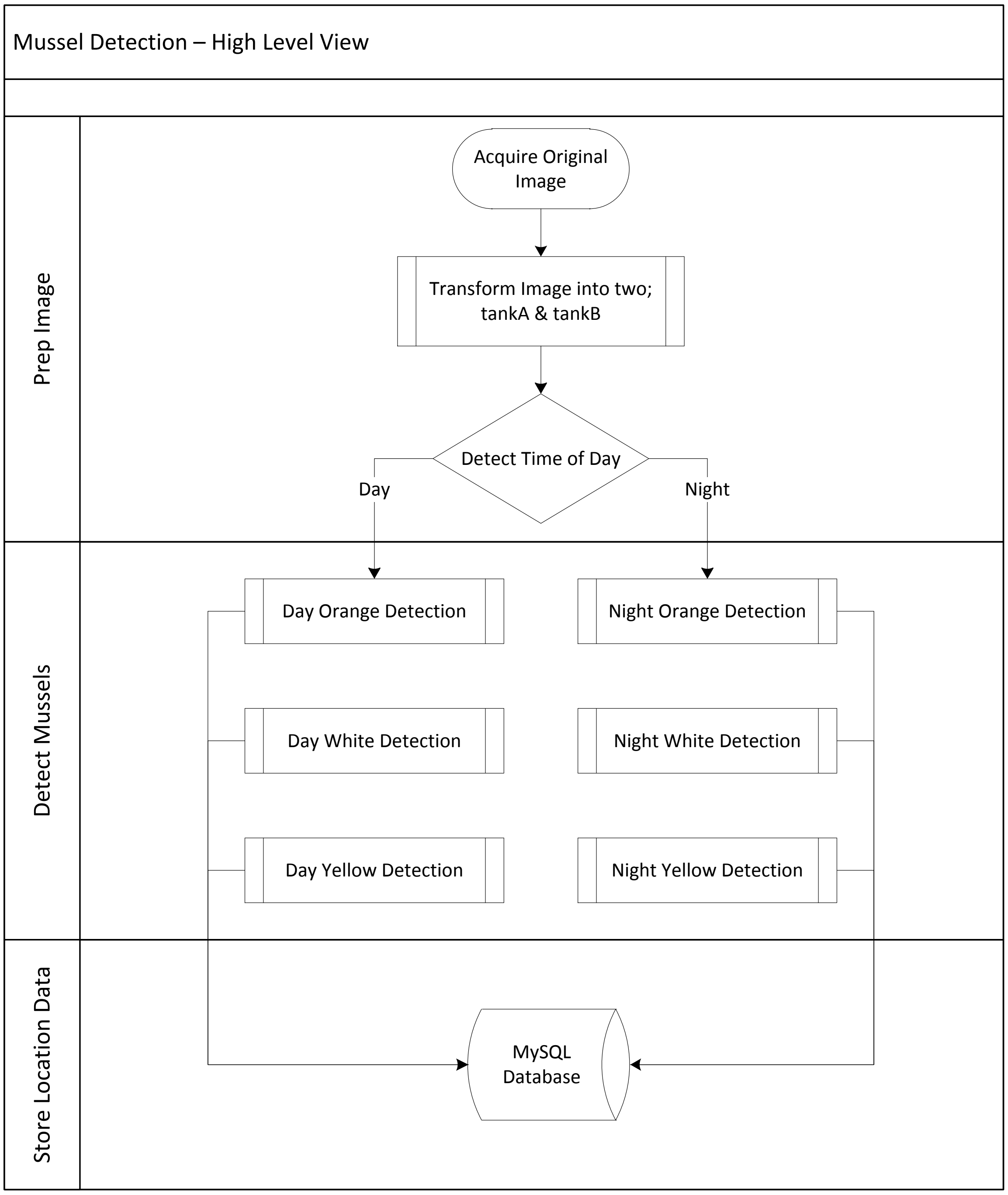


Figure 23 – Flowchart For Script Showing High Level View Of Mussel Detection.

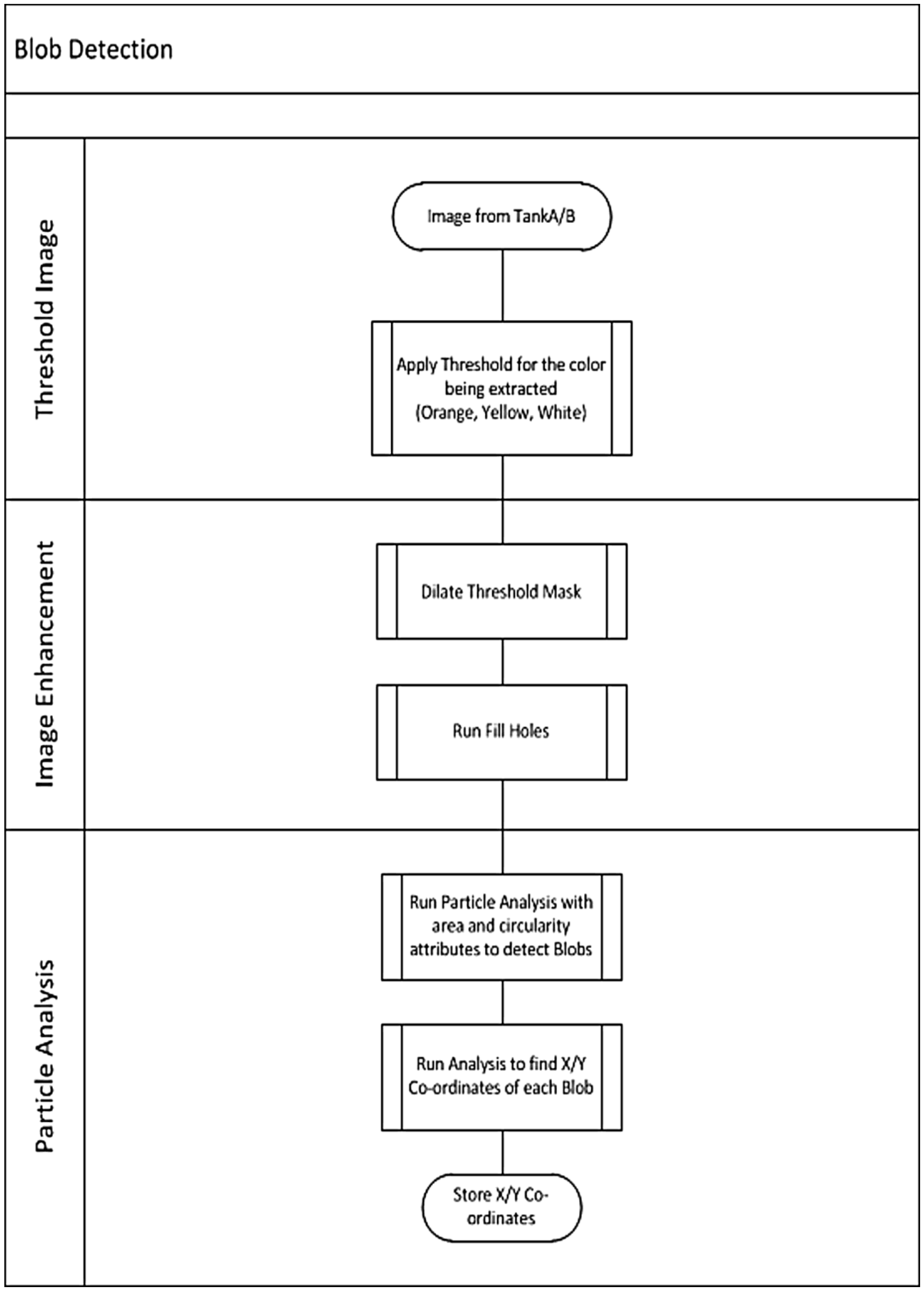

Blob Detection
Threshold Image
Image Enhancement
Particle Analysis
Image from TankA/B
Apply Threshold for the color being extracted (Orange, Yellow, White)
Dilate Threshold Mask
Run Fill Holes
Run Particle Analysis with area and circularity attributes to detect Blobs
Run Analysis to find X/Y Co-ordinates of each Blob
Store X/Y Co-ordinates

Figure 24 – Flow Chart Showing How to Extract Centroid Information from Each Color.

Further, in some cases, as shown in the mask in figure, the object of interest is not really of any uniform shape. There is a need to expand the pixels out, this will cause the nearby pixels to start touching each other and merge. We use the dilate function (Ferreira & Rasband, 26.8 Binary, 2011) to do this. To add another layer of robustness, we use the "fill in holes function" (Ferreira & Rasband, 26.8.8 Fill Holes, 2011) which will fill in the empty zones of enclosed blobs.

The next step is to analyze the segmented objects after thresholding of the images. We use the "analyze particles function" of ImageJ (ImageJ, 2012) to process the data to detect and filter on the objects of interest. We would like to filter non-sense data like objects that are on the absolute edges of the image, artifacts that have made it through color thresholding that are either too small or too big, etc. We achieve this by defining area, circularity parameters and excluding objects from out analysis that are touching the edges of the image by configuring the particle analysis input parameters. It is not plausible for a mussel to locate itself in such a manner that the painted region is touching the absolute edges of the tank, there is always going to be some sediment, in addition to this and the post-processing that transforms the picture with both tanks to its respective tanks assures that small portion of the walls of each tank are included in the image. Hence we would like to filter out what threshold algorithms can pick up. These can also be things like plumbing tubes and make their way into the field of view to show up in the image. Since there is a chance that these objects can be the same color as one of the 3 colors we are tracking, it will show up in the threshold masks to be processed. We disregard this kind of information from our analysis.

Figure 26 through Figure 30 below show incremental steps throughout the image processing pipeline for detection of mussels. It is clear that one can detect areas that are of no interest to us. Using the "analyze particles" function one can observe that we can reasonably detect only the mussels we want to track. We determined maxima and minima by applying the pipeline on a large data set and analyzing the output of area and circularity of each detected region of interest (ROI). With this approach, we have high confidence that this will yield an acceptable rate of success for detection.

Now each region of interest can be examined to find its centroid information, this information is extracted and saved into a table. Each ROI includes *Area*, *X-Coordinates*, *Y-Coordinates*, *Color*, Automatic *timestamp* of when it was captured, and the tank it came from. This will be made available via a MySQL library, making it very easy for scientist to gather data, at a certain time, color, from a specific tank, etc.

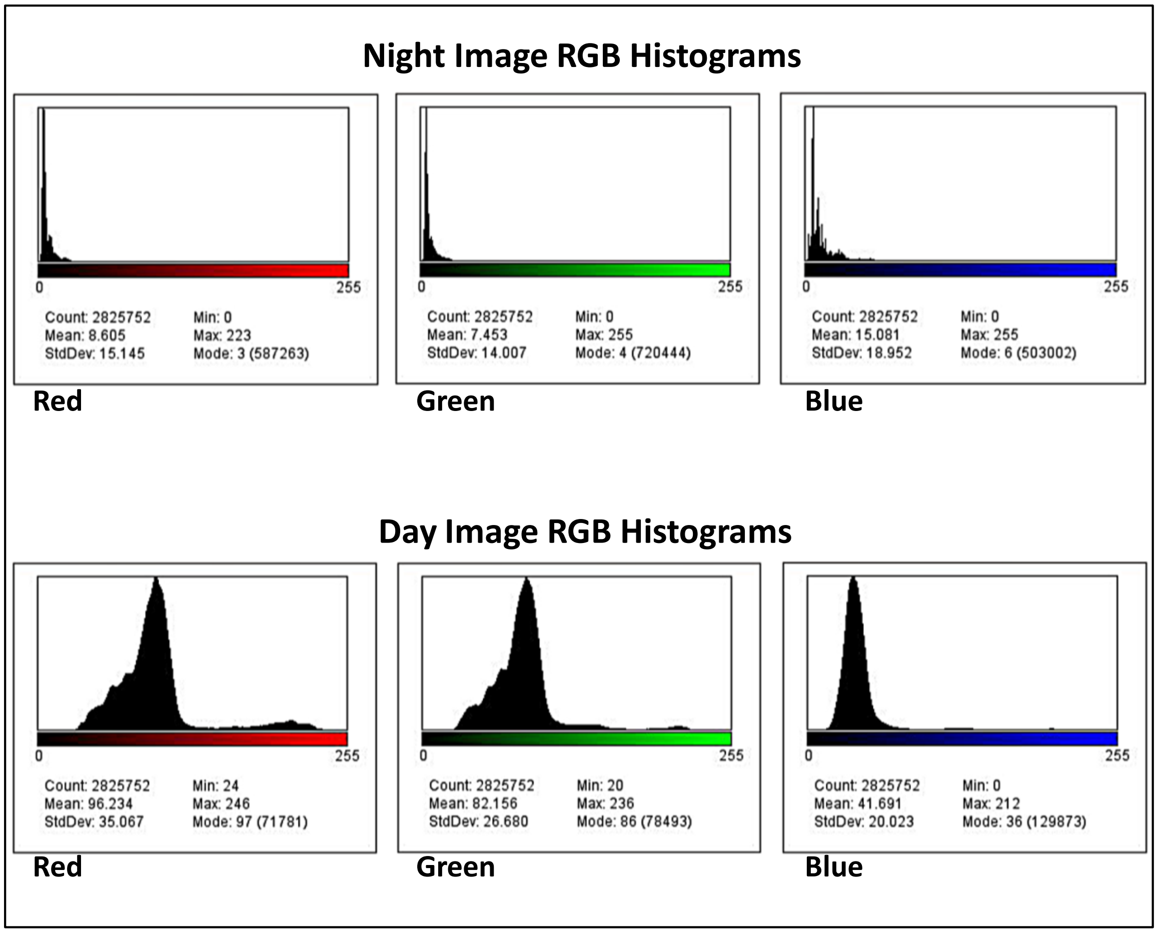


Figure 25 – Night RGB Histograms Shown On Top And Day At The Bottom.

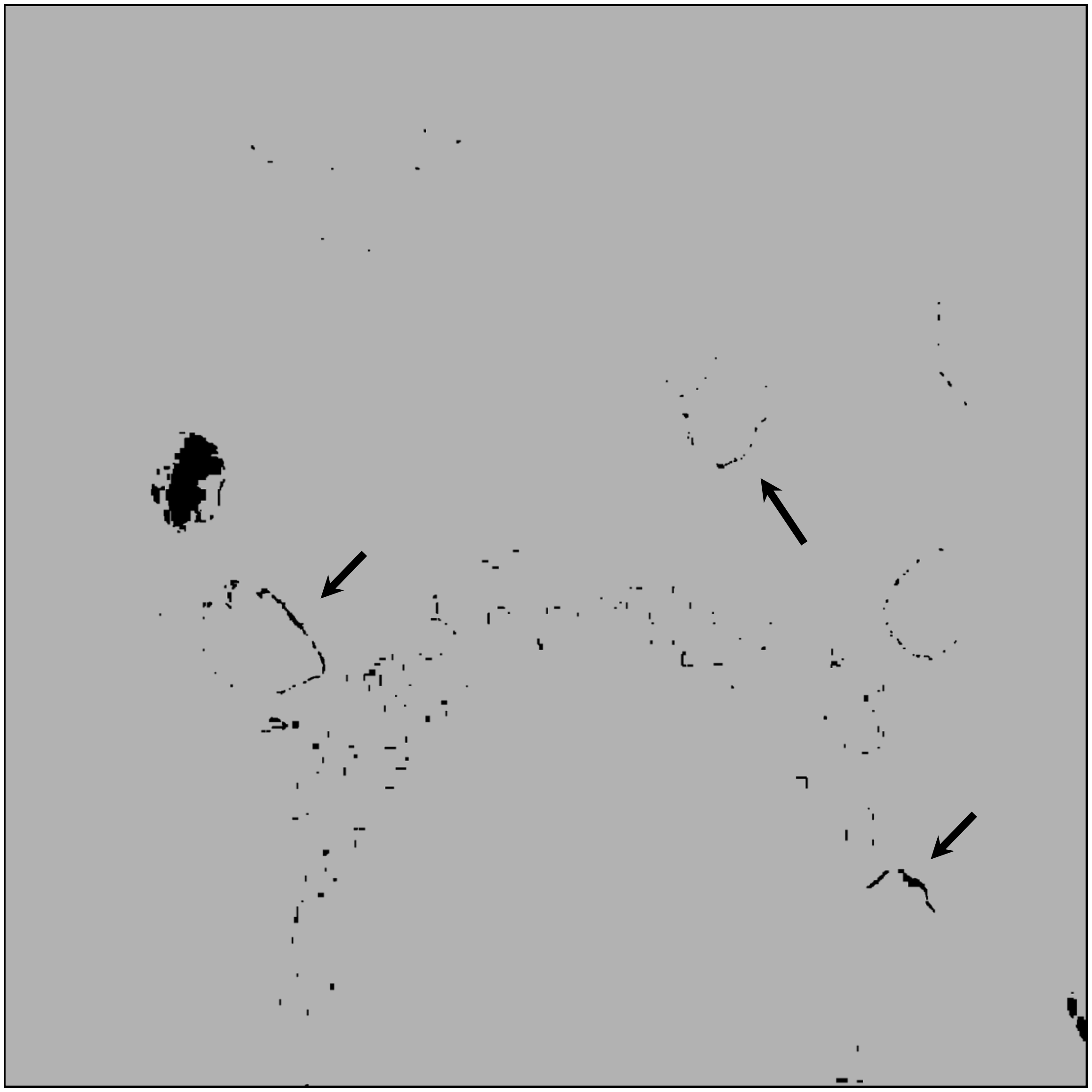


Figure 26 – Threshold Mask For Night Time White Shell Detection With Arrows Pointing Out The Unwanted Artifacts That Are Detected.

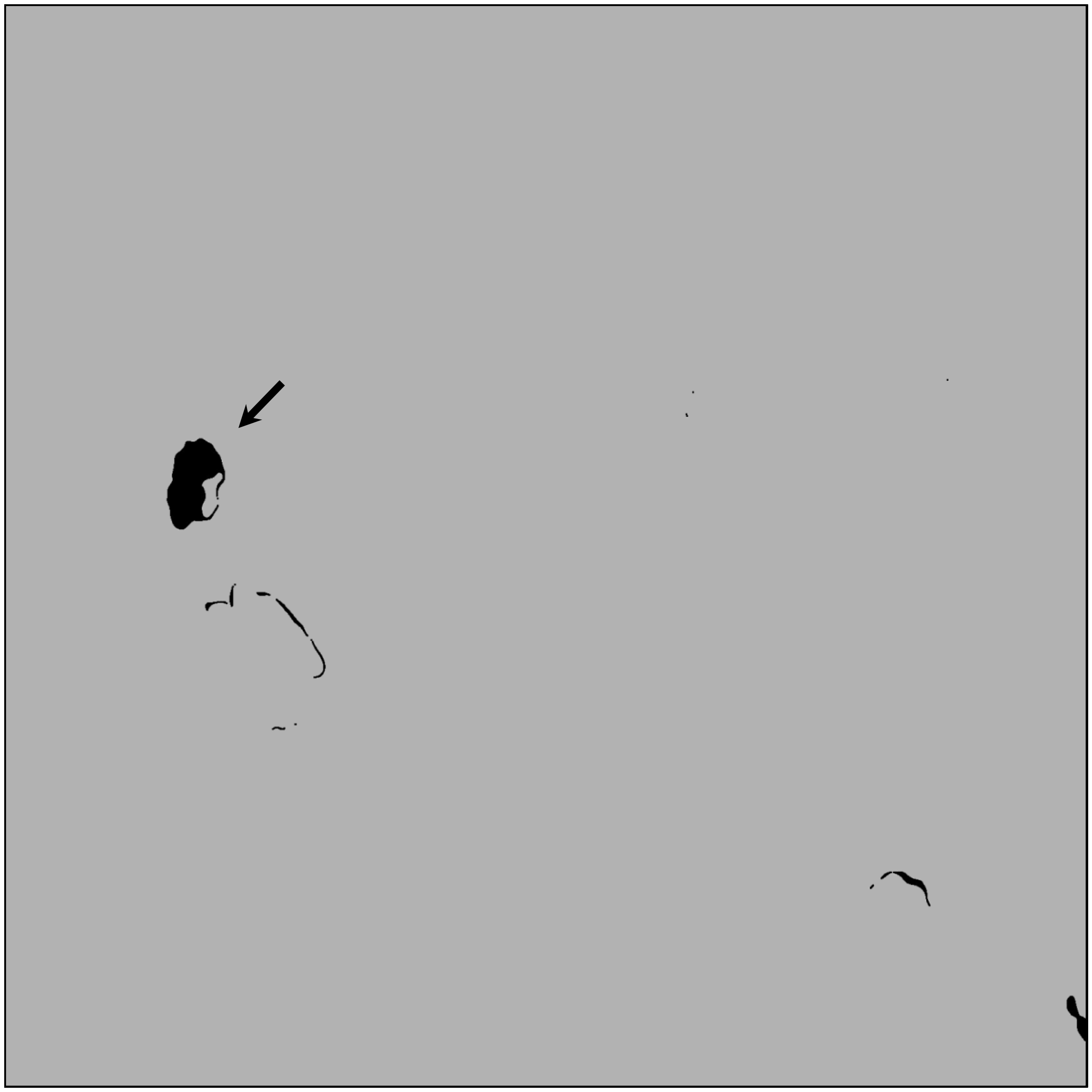


Figure 27 – White Shell Mask After Applying Dilate And Fill Holes Function With Arrow Pointing At The Detected White Shell.

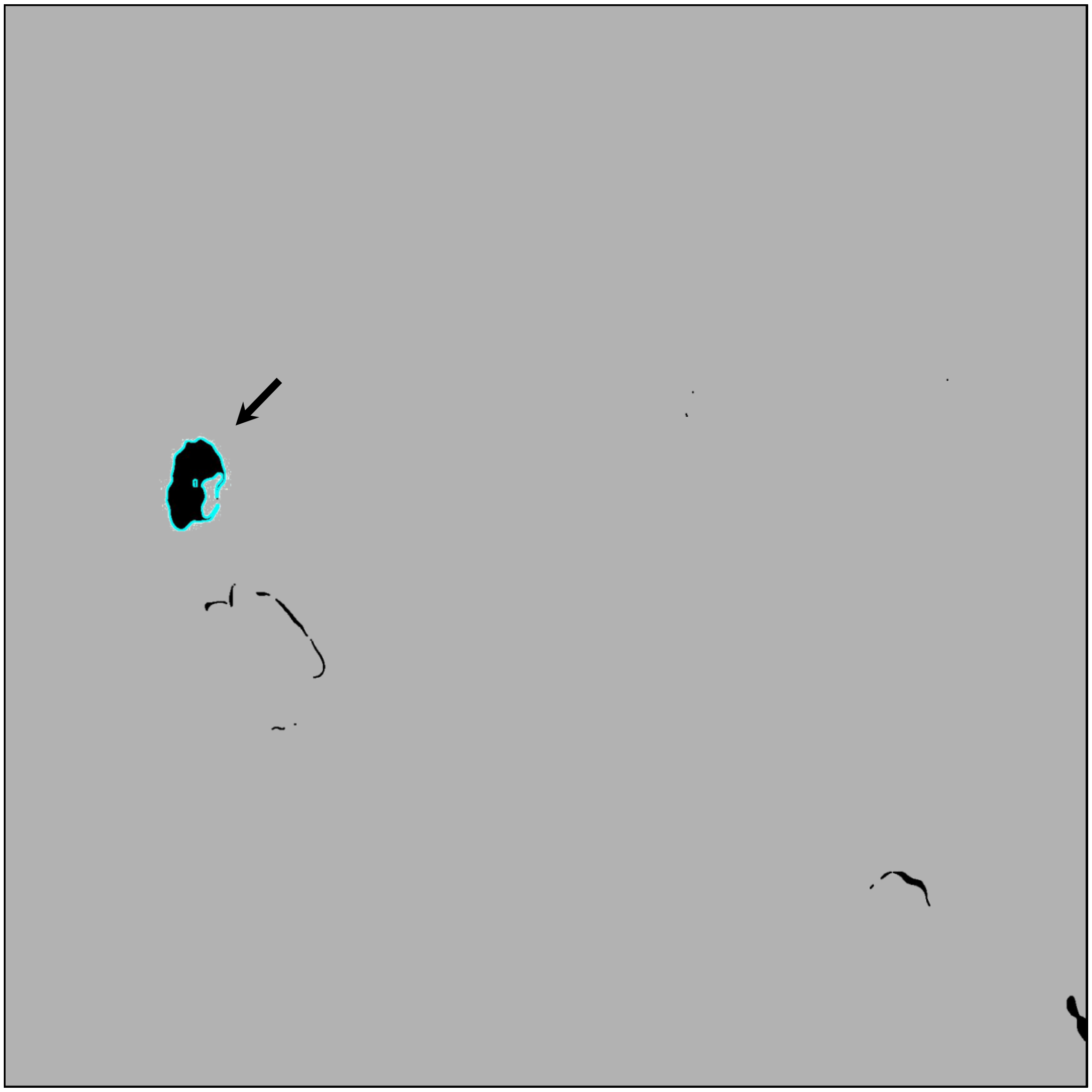


Figure 28 – Showing Particle Analysis With Parameter Adjustments: Only the Mussel Is Detected.

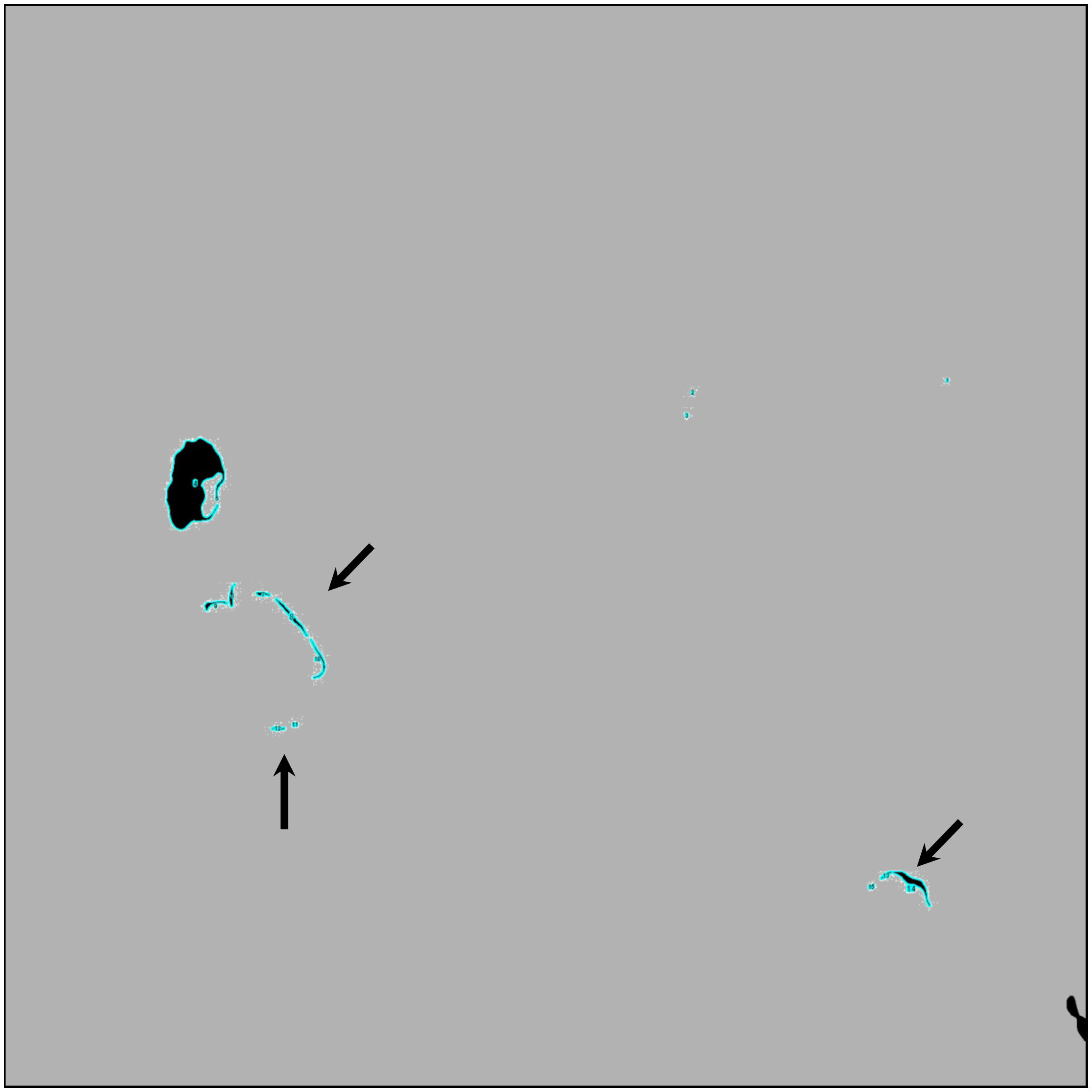


Figure 29 – Showing Particle Analysis Without Area And Circularity Filtering: Mussel And Other Artifacts Are Detected As Pointed Out By Arrows.

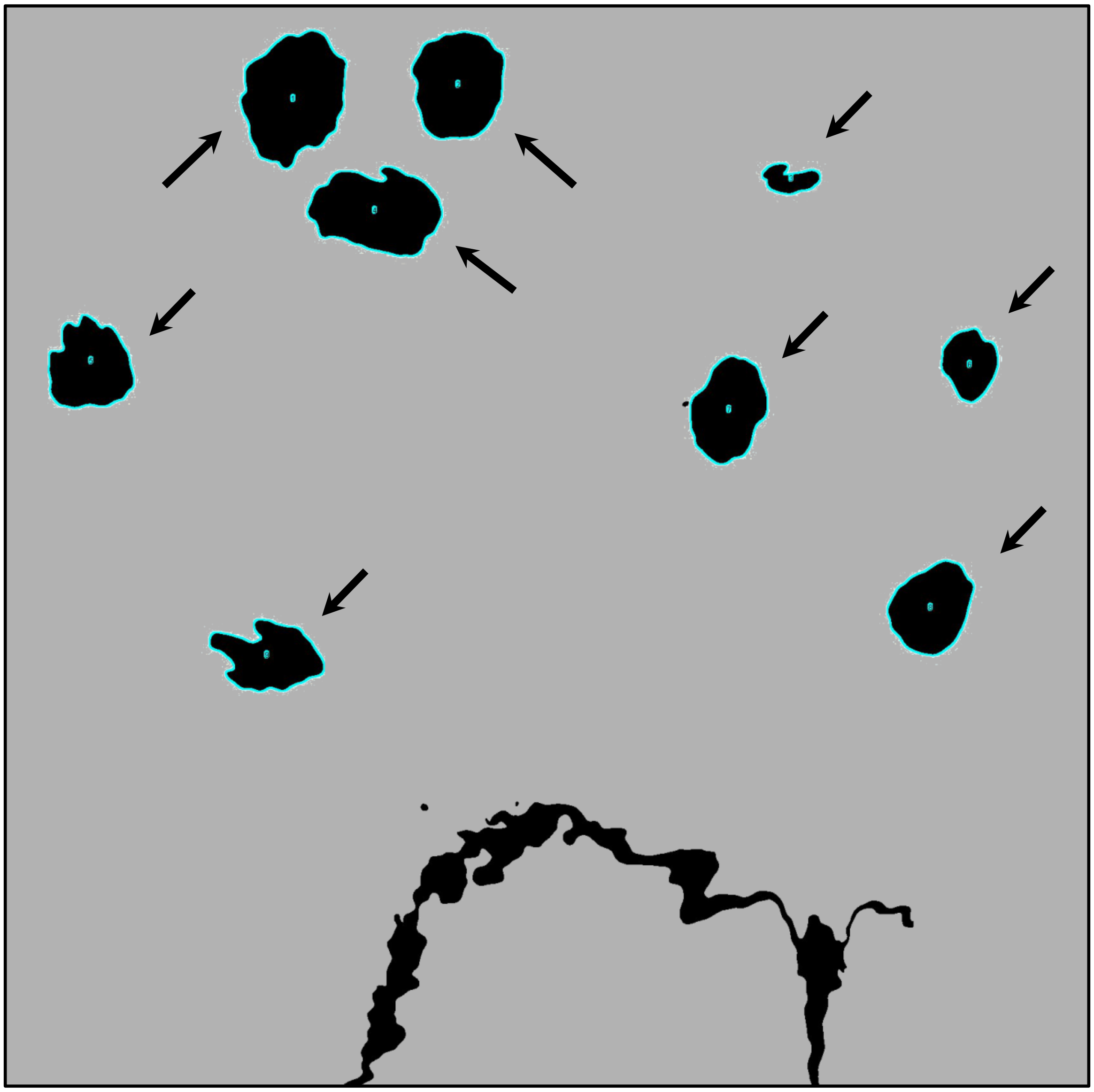


Figure 30 – Showing The Detected Mussels For Particle Analysis On Orange Shells At Night By Arrows.

In order to make the algorithm more robust to overcome situation when there can be overcrowding, we would like to be able to separate the individual shells. We experimented with so-called watershed methods. Watershed method finds the center of each object using a morphological erode operation, then calculates a distance map from the object center points to the edges of the objects, then fills that area with imaginary water. Where two watersheds meet, a line is drawn between them to split them. We could do these steps manually, but the “watershed” function in FIJI automates the process. But the problem with this method is that some individual uncluttered mussel shells that have uneven masks will also get split into multiple segments as shown in Figure 31. This causes detection discrepancies. In addition, there could be instances where watershed can create more blobs from unwanted artifacts that will pass the defined filters of blob detection as shown in Figure 32. We have also observed instances where single mussel is split into smaller pieces that will be rejected by the filter because meet the minimum size requirements.

Watershed method seems to work well for images where clustering of objects are in real high concentration, like in cell nuclei detection. Where the cells are in real high concentration and a vast majority of them are touching (Gonzalez & Woods, 2002). This is an unrealistic setup that would not be applicable to our project. To fix the cluttering problem we run particle analysis again with the area parameter adjusted to let between 2 and 3 mussels at most. Thus, we can determine out if there are instances where mussels are touching each other and log them at the same location into the database. This data can be extracted later to check if there was clustering behavior demonstrated by the mussels. Figure 24 shows the image processing pipeline.

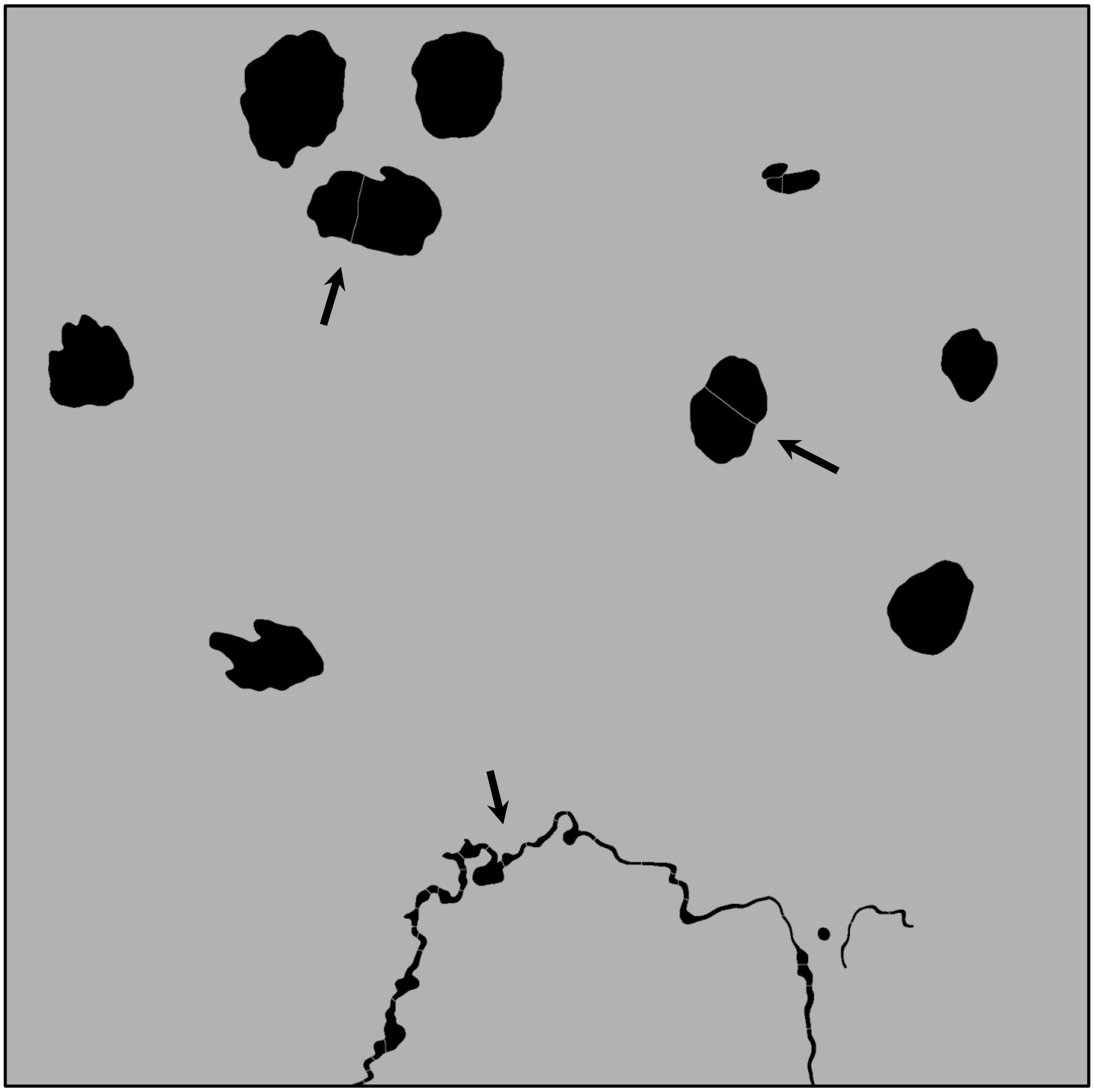


Figure 31 – Mask Of Image With Watershed Algorithm Applied; Mussels Are Segmented Into Multiples Pointed Out By Arrows.

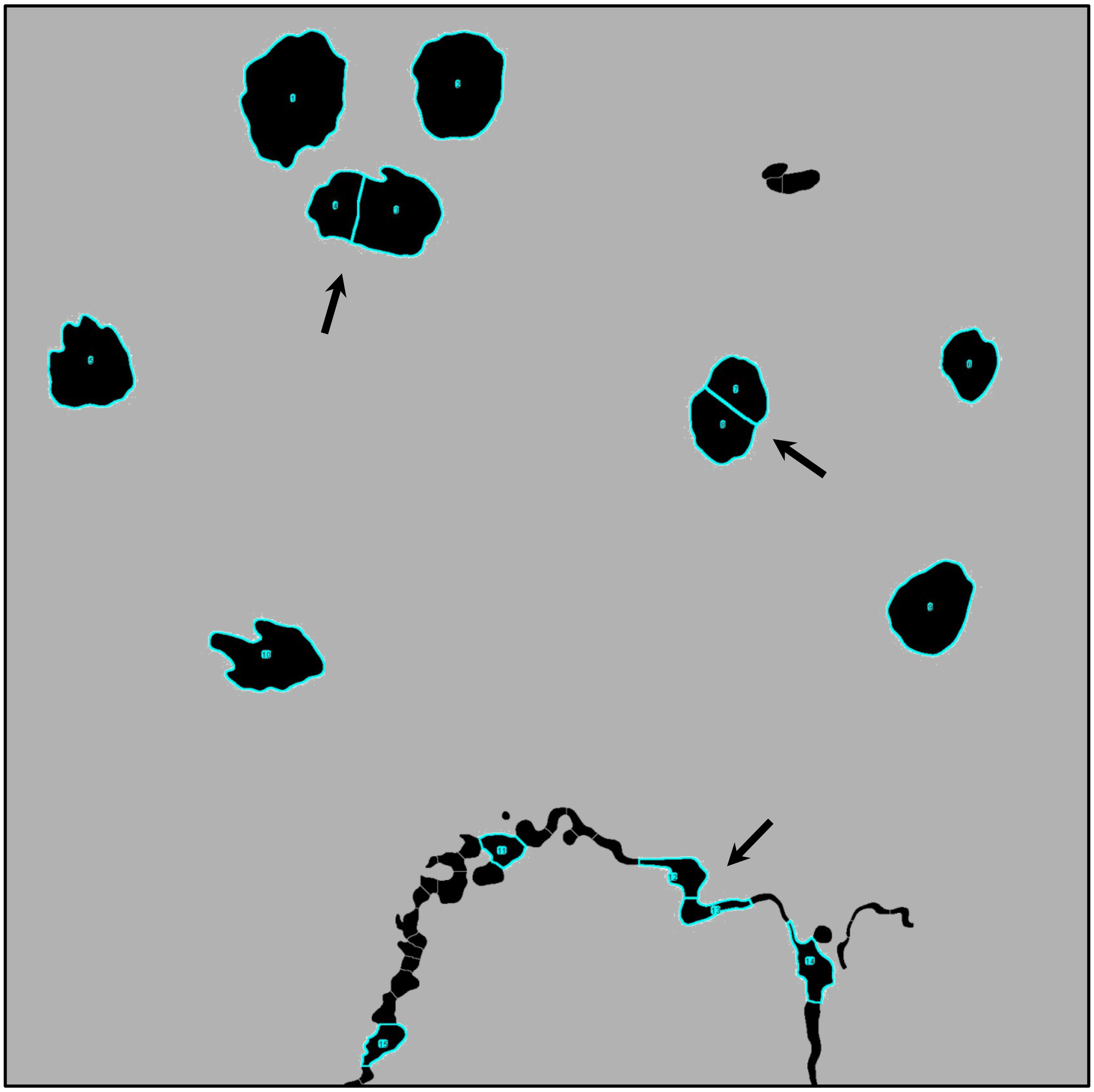


Figure 32 – Particle Analysis To Mask With Watershed Applied: Segmented Artifacts Detected as Mussels As Pointed Out By Arrows.

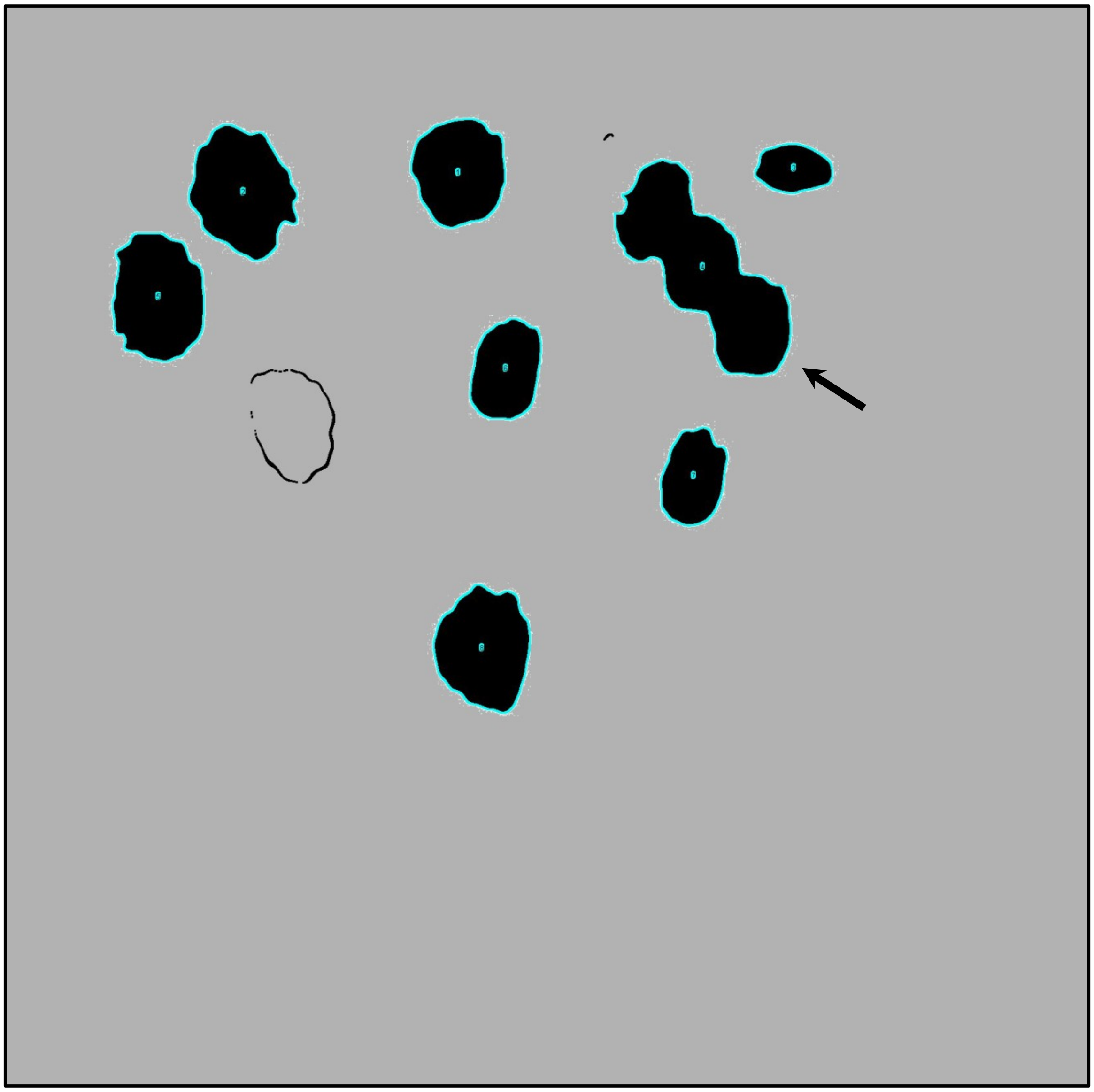


Figure 33 – Mask of A Clustered Mussel Picture, It Could Be Detected As A Single Mussel.

DESIGN OF SYSTEM

The purpose of this chapter is to outline the overall approach to designing a solution from scratch. The main goal was to build a platform that is highly scalable, component-based, and reusable and provide capability use any single component for rapid prototyping.

The overall design is modular rather than a single, complex, executable. Each module was designed to be independent of the next. Each module's functionality is derived from functional components of the solution such as traverse control, camera control, image processing, and video generation. Each one of these functions could be used in other experiments, might need to be modernized for newer equipment upgrades, expanded for upcoming needs, migration of platform, etc. This approach increases reusability of the system as well as each individual module.

PYTHON is very powerful and popular scripting language. It is available on multiple platforms, does not need fancy development environment and compilers, and provides hooks to just about any application. More importantly, it's becoming widespread in the Software Engineering and Science community. This allows for quick turnaround time for proof of concepts, and reduces development time significantly.

Camera Control Module

For the controlling the camera we used Visual C++ using Visual Studio. The implementation is based off of sample code provided by Canon (Canon, 2012). We kept the existing code architecture because it had a good base of needed functionality and had sound design with built-in exception handling. We added code for file handling to save, move pictures taken, hooks for FTP client functionality to upload pictures to an offsite server. . We expanded the functionality to keep the camera alive the throughout the experiment without entering power save mode, and time lapse trigger. Having this component ready to be deployed and tested was important, because we were able to do some long term deployment to test the camera while we developed other components.

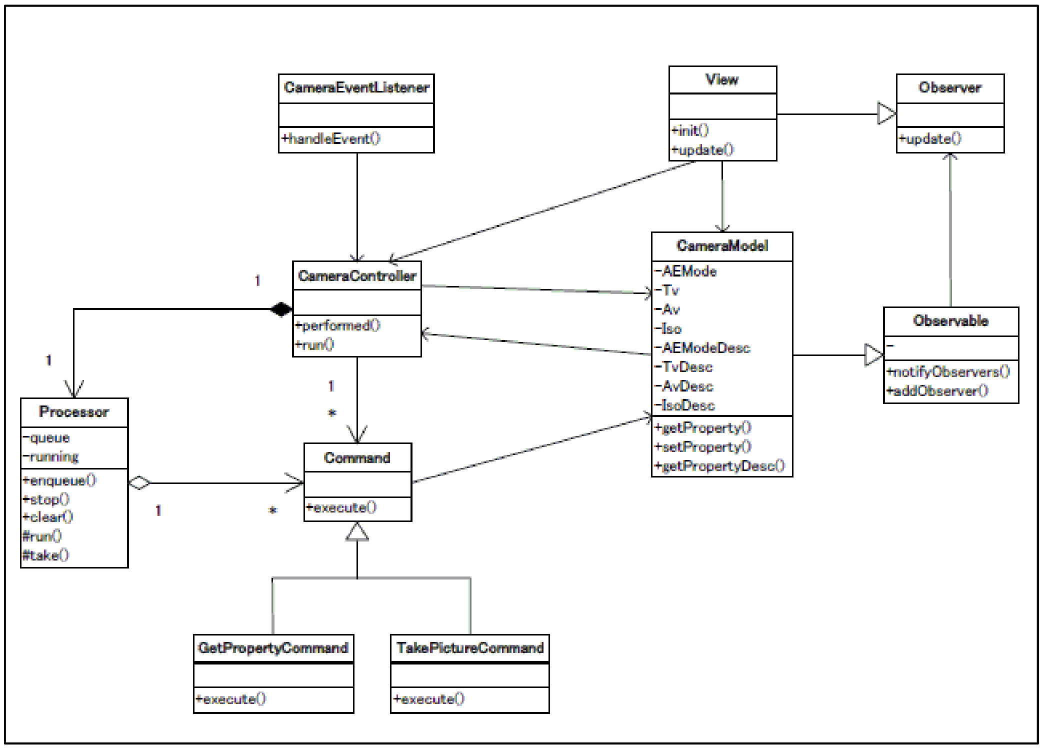


Figure 34 – Class Diagram For Camera Control Software.

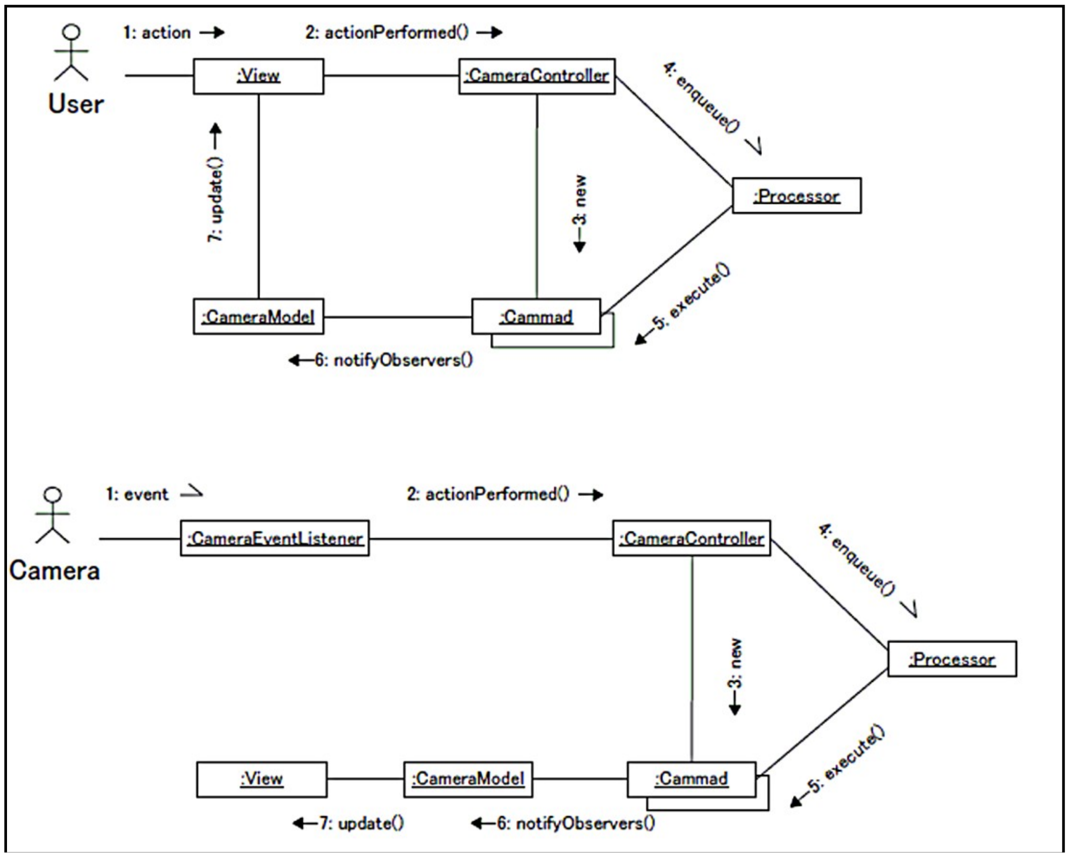
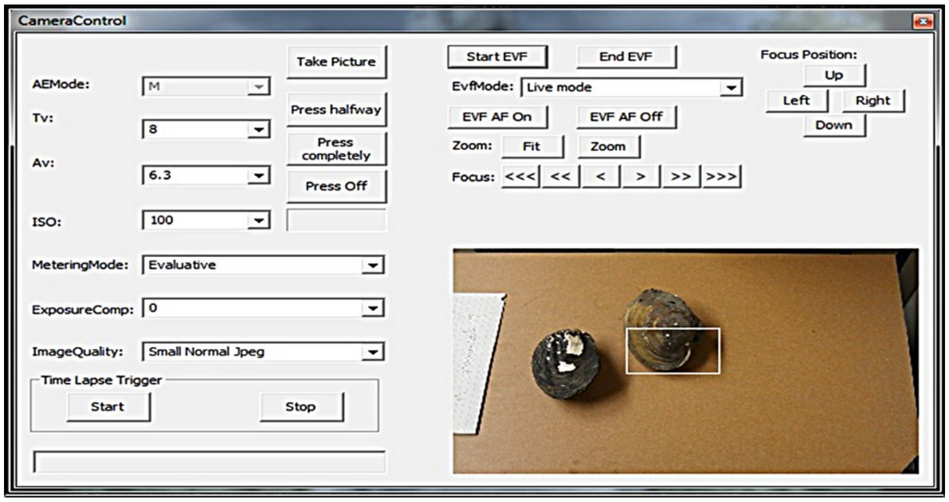


Figure 35 – Collaboration Chart for Camera Control Software.



AEMode: M
Tv: 8
Av: 6.3
ISO: 100
MeteringMode: Evaluative
ExposureComp: 0
ImageQuality: Small Normal Jpeg
EvfMode: Live mode

Figure 36 – Diagram Of Camera Control Interface.

Video Generation Module

To generate the videos we are using two different applications, mencoder from mplayer (MPlayer, 2011) and ffmpeg (FFmpeg, 2011). To generate video from images, we copy images to a temporary location on the local drive, post process them, and generate new video concatenated with the old ones. We use a PYTHON script efficiently performs all of these tasks, and invoke the applications with the required parameters.

Traverse Control Module

For traverse control we use a standalone PYTHON again. PYTHON provides easy access and integration to MySQL (MySQL, 2012) and serial ports (pySerial, 2010). Something that can take hundreds of lines of code can be easily written in PYTHON with the same functionality in a fraction of the lines of code. This makes is perfect for fast prototyping as well.

Image Processing Module

Image processing is usually challenging for developers that do not have comprehensive background on imaging and imaging techniques so it is important to have a way to explore how various algorithms and pipelines work with real life data. FIJI (FIJI, 2012) provides this means. It is user-friendly and the developer can see affects in real time without having to change code and recompile every time. We use PYTHON-based scripting through FIJI to do our image processing

Web Module

Web availability is provided via WAMP (Bourdon) that uses industry-standard PHP, MySQL, Apache, etc. that can be easily deployed even on a user desktop to test and develop new pages and verify additional functionality. The web applications were built to be independent of the rest of the modules, so this can be reused or expanded as needed without having to recompile the whole project, re-write any of the code.

REMOTE AVAILABILITY THROUGH THE WWW

The captured image, post-processed location data, and time lapse videos generated by the system are made available to the end-user via a webpage. A web server is needed to generate dynamic content for the pages, store and query location data and make available the stored pictures and videos throughout the length of the experiment. This also gives the flexibility to the research group to share their findings with offsite scientist by just providing a link to the website. For this project we chose WAMP. WAMP is a collection of packages of independently-created open-source programs installed on computers that use a Microsoft Windows operating system (Bourdon).

WAMP is an acronym formed from the initials of the operating system Microsoft Windows and the principal components of the package: Apache, MySQL and one of PHP. Apache is a web server. MySQL is an open-source database. PHP is a scripting language that can manipulate information held in a database and generate web pages dynamically each time content is requested by a browser. (Bourdon)

The flowcharts in Figure 37 and Figure 38 show how the each of the main components of the server interacts to generate the web pages.

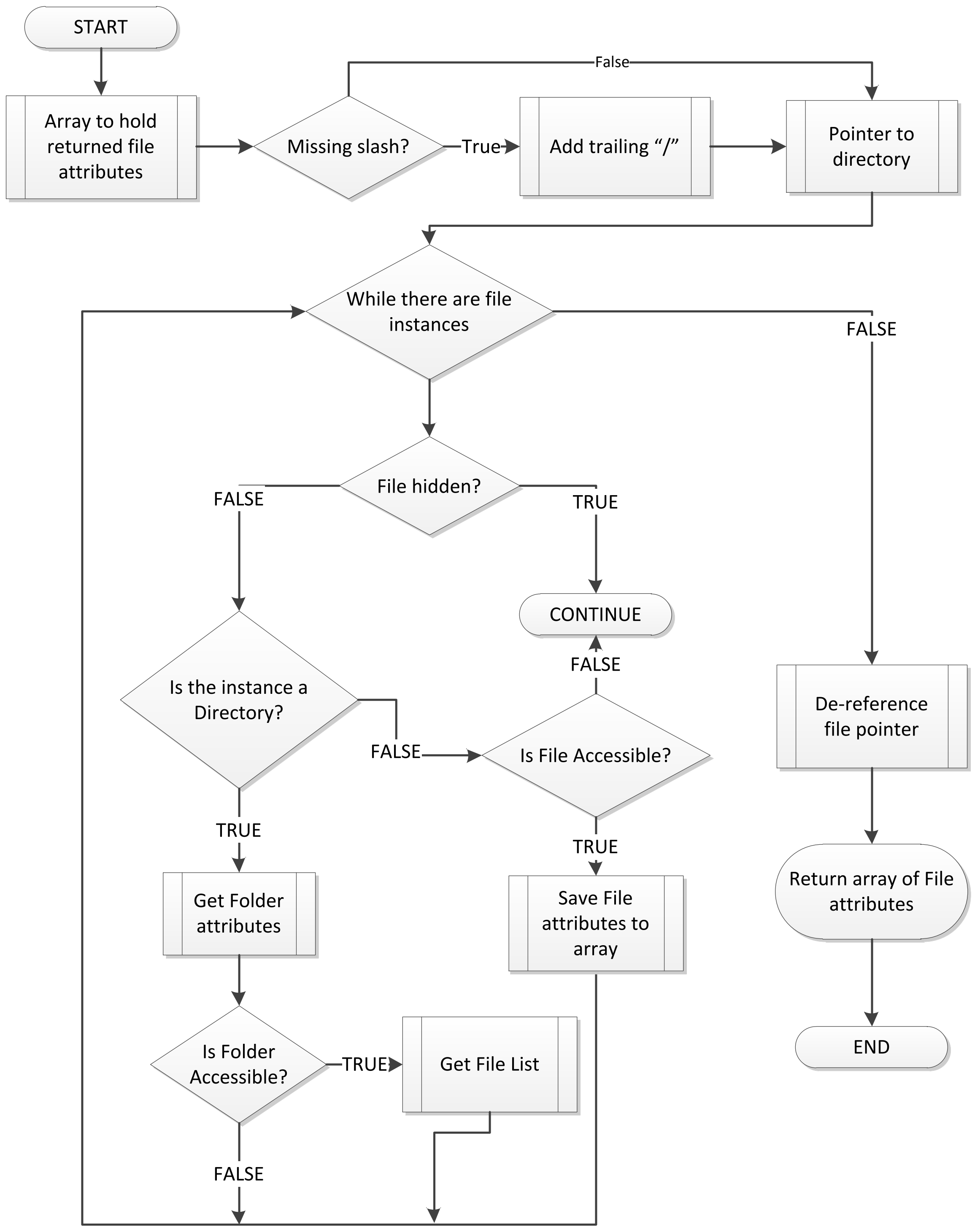


Figure 37 – Flowchart For Function That Generates List Of Files In Directory.

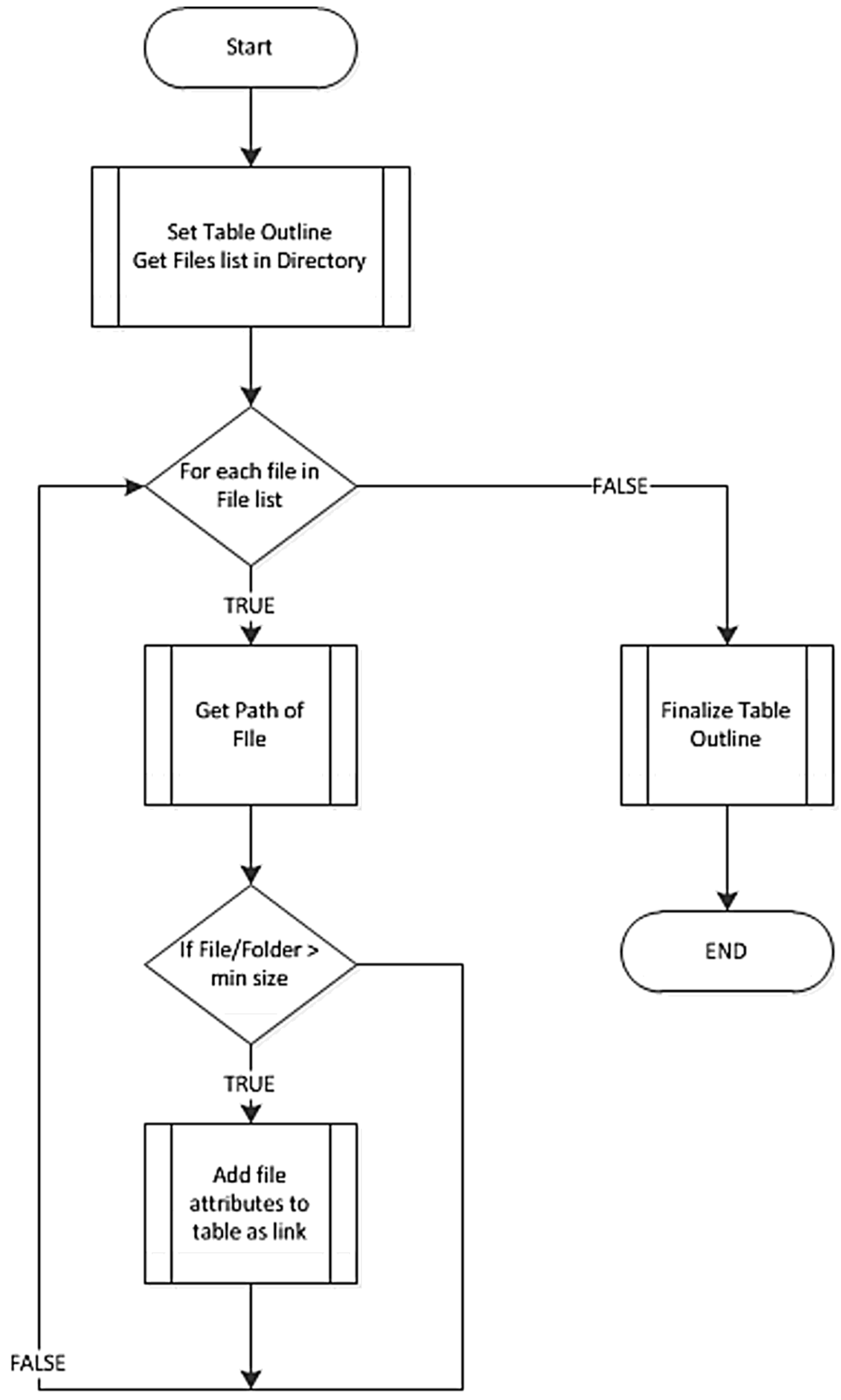

Start
Set Table Outline
Get Files list in Directory
For each file in File list
FALSE
TRUE
Get Path of FIle
Finalize Table Outline
END
If File/Folder > min size
TRUE
Add file attributes to table as link
FALSE

Figure 38 – Flowchart Of PHP Script That Generates Directory Listing Page.

For ease of use and navigation there are three main PHP pages for this project namely the main page, the image directory and the query page. The main page serves as a welcome page. We have a short introduction and the latest image taken from the camera for the day. There are links to the images and videos generated. Below is a snapshot of this page along with the PHP design. The second PHP page traverses through all the images and videos that are available since the beginning of the project, and makes them available to user via links so that, if need be the end-user can check each individual image taken at a certain time.

The last page is the query page, where the data in the database for the location data is stored is made available to the user to search and generate data sets that can be used to study the behavior of the mussels. The user input criteria are based on how the data is stored in the MySQL database; this is covered in the MySQL section. Image below shows a snapshot of this page with an example data set that was generated from the input of the user.

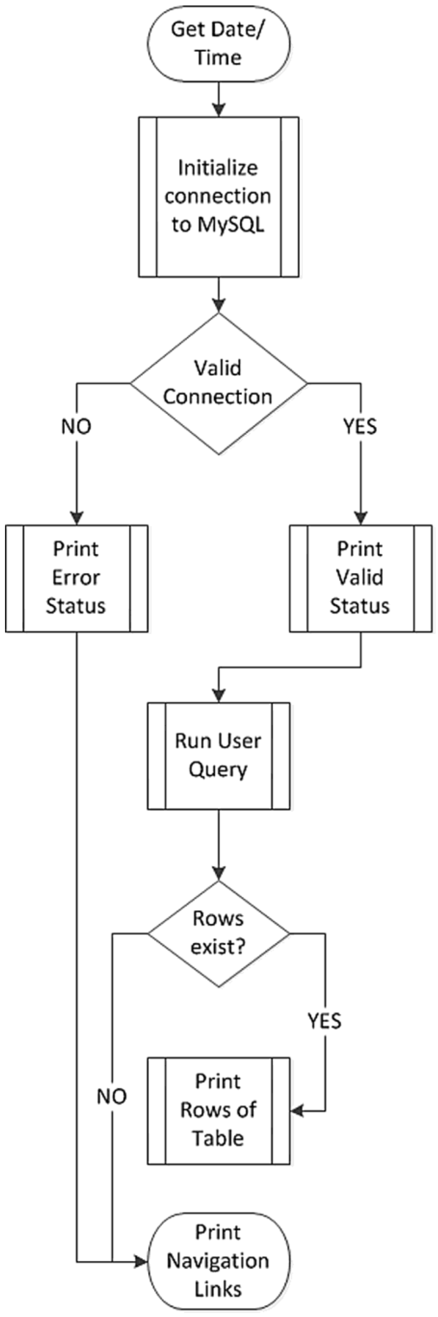
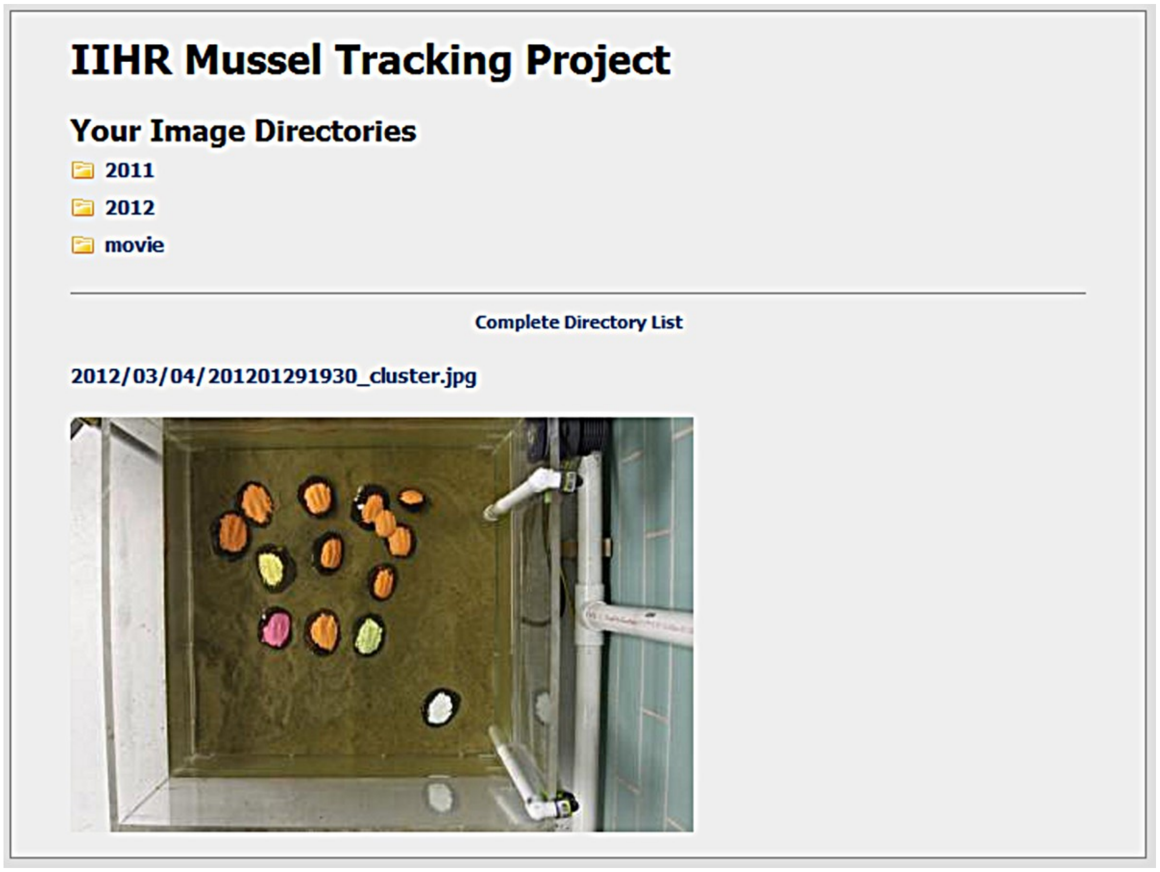


Figure 39 – Flowchart For PHP Script That Generates MySQL Page.



IIHR Mussel Tracking Project

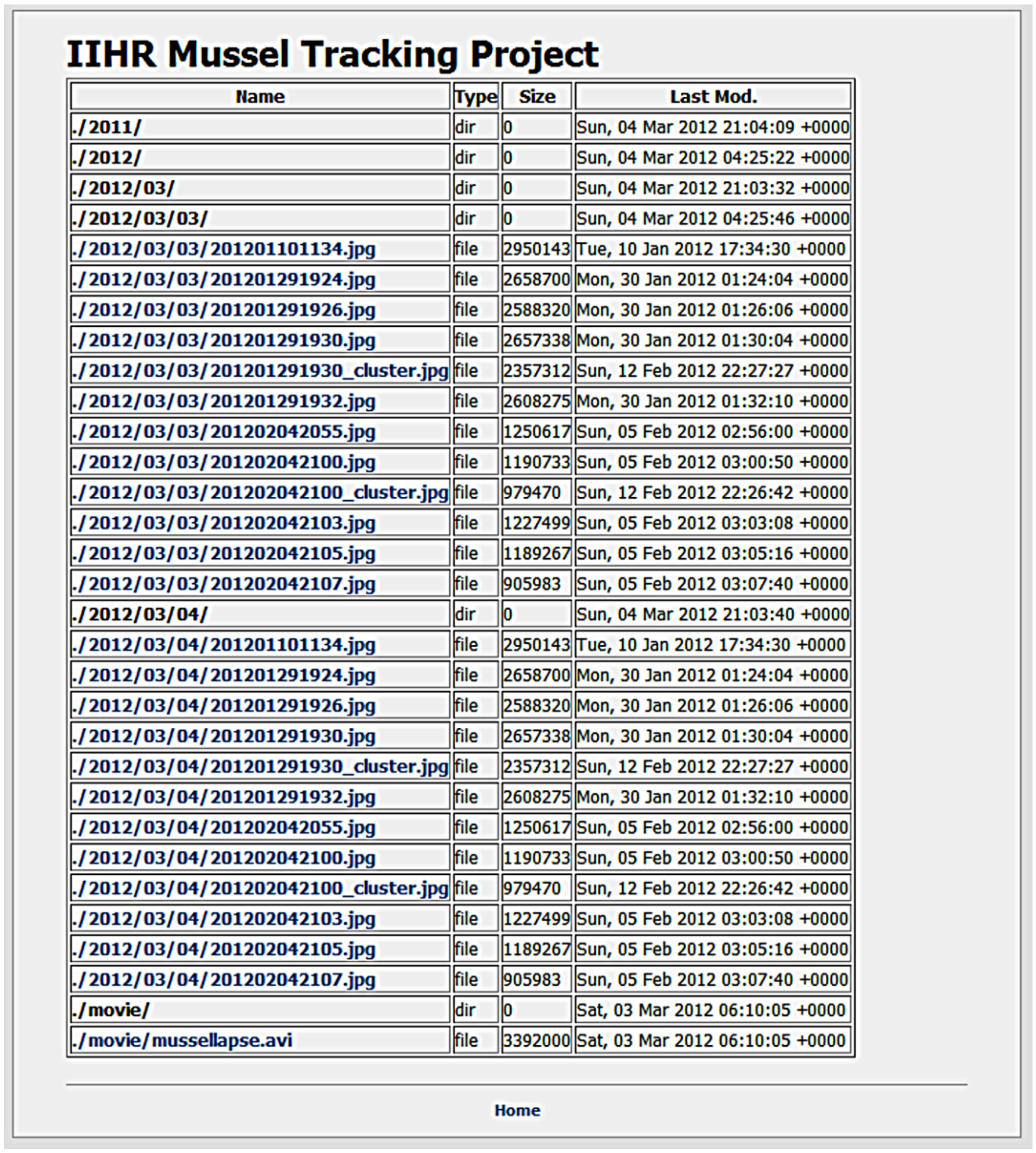
Your Image Directories

- 2011
- 2012
- movie

Complete Directory List

2012/03/04/201201291930_cluster.jpg

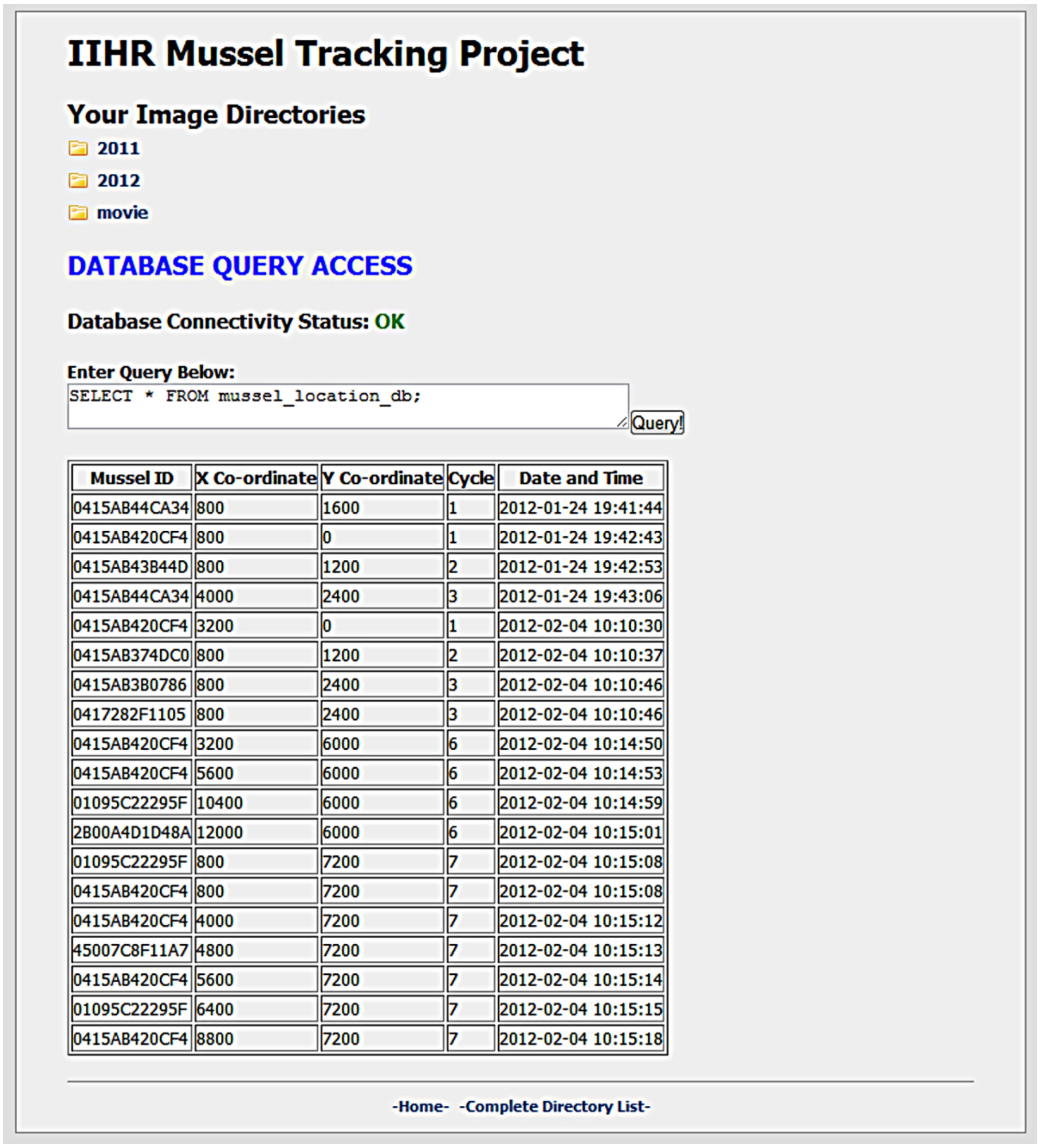
Figure 40 – Image Of The IIHR Mussel Tracking Index Page.



IIHR Mussel Tracking Project

Name	Type	Size	Last Mod.
./2011/	dir	0	Sun, 04 Mar 2012 21:04:09 +0000
./2012/	dir	0	Sun, 04 Mar 2012 04:25:22 +0000
./2012/03/	dir	0	Sun, 04 Mar 2012 21:03:32 +0000
./2012/03/03/	dir	0	Sun, 04 Mar 2012 04:25:46 +0000
./2012/03/03/201201101134.jpg	file	2950143	Tue, 10 Jan 2012 17:34:30 +0000
./2012/03/03/201201291924.jpg	file	2658700	Mon, 30 Jan 2012 01:24:04 +0000
./2012/03/03/201201291926.jpg	file	2588320	Mon, 30 Jan 2012 01:26:06 +0000
./2012/03/03/201201291930.jpg	file	2657338	Mon, 30 Jan 2012 01:30:04 +0000
./2012/03/03/201201291930_cluster.jpg	file	2357312	Sun, 12 Feb 2012 22:27:27 +0000
./2012/03/03/201201291932.jpg	file	2608275	Mon, 30 Jan 2012 01:32:10 +0000
./2012/03/03/201202042055.jpg	file	1250617	Sun, 05 Feb 2012 02:56:00 +0000
./2012/03/03/201202042100.jpg	file	1190733	Sun, 05 Feb 2012 03:00:50 +0000
./2012/03/03/201202042100_cluster.jpg	file	979470	Sun, 12 Feb 2012 22:26:42 +0000
./2012/03/03/201202042103.jpg	file	1227499	Sun, 05 Feb 2012 03:03:08 +0000
./2012/03/03/201202042105.jpg	file	1189267	Sun, 05 Feb 2012 03:05:16 +0000
./2012/03/03/201202042107.jpg	file	905983	Sun, 05 Feb 2012 03:07:40 +0000
./2012/03/04/	dir	0	Sun, 04 Mar 2012 21:03:40 +0000
./2012/03/04/201201101134.jpg	file	2950143	Tue, 10 Jan 2012 17:34:30 +0000
./2012/03/04/201201291924.jpg	file	2658700	Mon, 30 Jan 2012 01:24:04 +0000
./2012/03/04/201201291926.jpg	file	2588320	Mon, 30 Jan 2012 01:26:06 +0000
./2012/03/04/201201291930.jpg	file	2657338	Mon, 30 Jan 2012 01:30:04 +0000
./2012/03/04/201201291930_cluster.jpg	file	2357312	Sun, 12 Feb 2012 22:27:27 +0000
./2012/03/04/201201291932.jpg	file	2608275	Mon, 30 Jan 2012 01:32:10 +0000
./2012/03/04/201202042055.jpg	file	1250617	Sun, 05 Feb 2012 02:56:00 +0000
./2012/03/04/201202042100.jpg	file	1190733	Sun, 05 Feb 2012 03:00:50 +0000
./2012/03/04/201202042100_cluster.jpg	file	979470	Sun, 12 Feb 2012 22:26:42 +0000
./2012/03/04/201202042103.jpg	file	1227499	Sun, 05 Feb 2012 03:03:08 +0000
./2012/03/04/201202042105.jpg	file	1189267	Sun, 05 Feb 2012 03:05:16 +0000
./2012/03/04/201202042107.jpg	file	905983	Sun, 05 Feb 2012 03:07:40 +0000
./movie/	dir	0	Sat, 03 Mar 2012 06:10:05 +0000
./movie/mussellapse.avi	file	3392000	Sat, 03 Mar 2012 06:10:05 +0000

Figure 41 – Image of The IIHR Mussel Tracking Directory List Page.



IIHR Mussel Tracking Project

Your Image Directories

📁 2011

📁 2012

📁 movie

DATABASE QUERY ACCESS

Database Connectivity Status: OK

Enter Query Below:

```
SELECT * FROM mussel_location_db;
```

Mussel ID	X Co-ordinate	Y Co-ordinate	Cycle	Date and Time
0415AB44CA34	800	1600	1	2012-01-24 19:41:44
0415AB420CF4	800	0	1	2012-01-24 19:42:43
0415AB43B44D	800	1200	2	2012-01-24 19:42:53
0415AB44CA34	4000	2400	3	2012-01-24 19:43:06
0415AB420CF4	3200	0	1	2012-02-04 10:10:30
0415AB374DC0	800	1200	2	2012-02-04 10:10:37
0415AB3B0786	800	2400	3	2012-02-04 10:10:46
0417282F1105	800	2400	3	2012-02-04 10:10:46
0415AB420CF4	3200	6000	6	2012-02-04 10:14:50
0415AB420CF4	5600	6000	6	2012-02-04 10:14:53
01095C22295F	10400	6000	6	2012-02-04 10:14:59
2B00A4D1D48A	12000	6000	6	2012-02-04 10:15:01
01095C22295F	800	7200	7	2012-02-04 10:15:08
0415AB420CF4	800	7200	7	2012-02-04 10:15:08
0415AB420CF4	4000	7200	7	2012-02-04 10:15:12
45007C8F11A7	4800	7200	7	2012-02-04 10:15:13
0415AB420CF4	5600	7200	7	2012-02-04 10:15:14
01095C22295F	6400	7200	7	2012-02-04 10:15:15
0415AB420CF4	8800	7200	7	2012-02-04 10:15:18

Figure 42 – Image of IIHR Mussel Tracking Database Access Page.

TIME-LAPSE VIDEO GENERATION

At the end of each day a script runs to automatically process the images generated by the camera. This is achieved by a multiple step process that utilizes functionality provided natively by PYTHON and two other applications that are used to create the videos. The daily pictures are saved in a web folder arranged by year, month and day, the image names are based on the current date and time and are in the format of YYYYMMDDhhmm.jpg where YYYY is the year, MM the moth, DD the day, hh the hour in 24 hour format and mm are the minutes. The folder structure is as shown in Figure 43 below

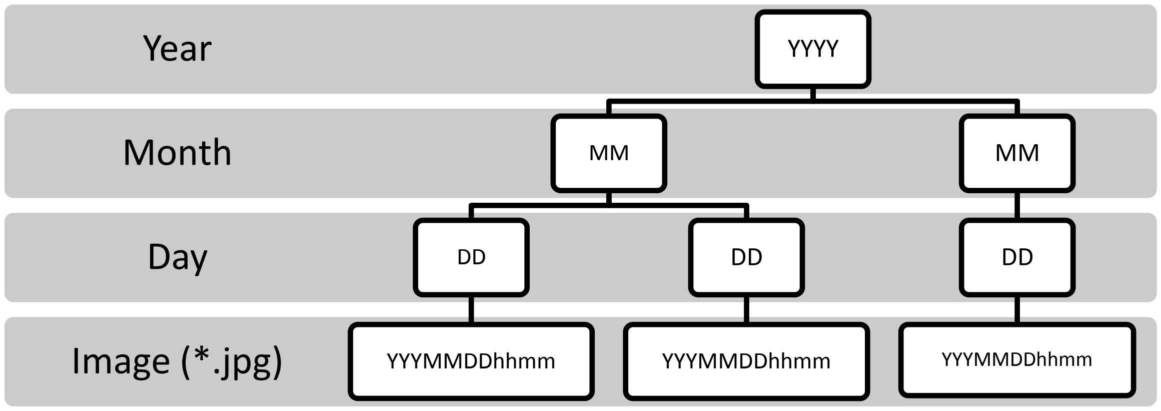


Figure 43 – Flow Diagram Of The Picture Naming Convention.

The other two applications used are open source highly optimized applications for rapid video generation. First application is called ffmpeg (FFmpeg, 2011), we use this application to generate a daily video from a folder of the day, the second is called Mplayer (MPlayer, 2011), and we use the mencoder (MPlayer) application to stitch the daily video on to an accumulated video. This allows the user to daily check the movement, or watch the complete progress since the start of the project.

First, the files are moved to a temporary folder to be pre-processed to be passed in as inputs to ffmpeg. To achieve this, we have a few error-checking steps before we proceed any further, so we must first check if the folders for the day's videos exist. If so, then create the folder that will contain the files, if this folder exists already we delete it, this may be due to some unknown fault that caused a crash in the previous run. After these checks, the files are renamed from their original naming convention that is suitable for web viewing to what is appropriate for this application. The format that works best is using incremented indexing of format "name_%4.jpg" where %4 is representing a 4 digit number. PYTHON scripts takes each image in the folder in order and renames them as image_####.jpg.

The script than determines the name of the video to be generated. If the script is running for the first time and the initial vide is not present then the videos are named accordingly, these are referred to as base_video and day_video. Where the base_video refers to the cumulative video and the day_video is what corresponds to that day's video to be generated. These variables are fed into ffmpeg applications image to video function called "image2" to generate the video (FFMpeg, 2012).

The last step is to concatenate the base_video with the new that day's video. here too we first check to see if this is the first base video being created, then pass the base video and the daily video generated by "ffmpeg" into" mencoder" to generate a temp video file. The old base_video is deleted and the new file is renamed appropriately. The flowchart of operation by this script is shown in Figure 44 below.

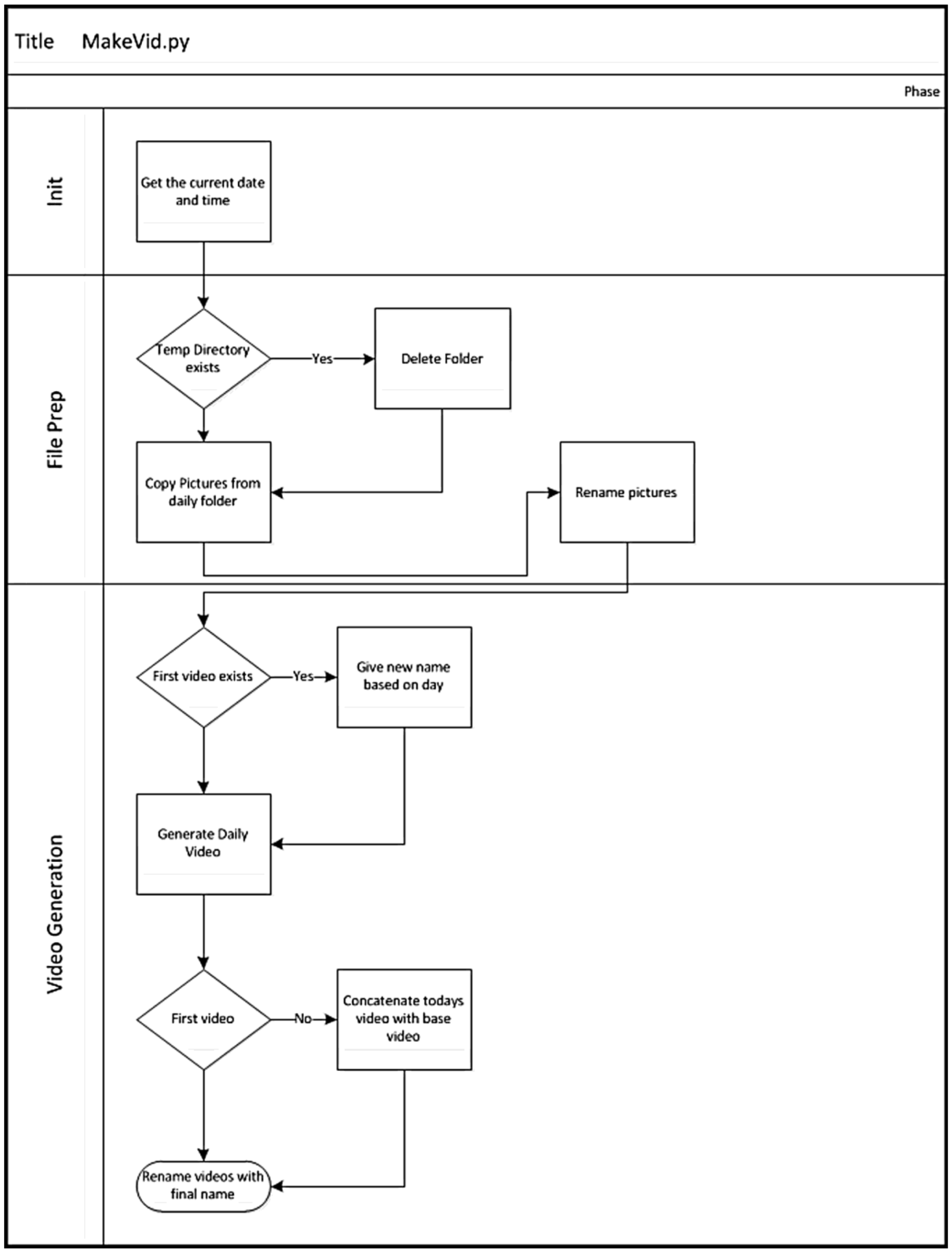


Figure 44 – Flowchart of Video Generation Script Written PYTHON.

STORING LOCATION DATA

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP web application software stack—LAMP is an acronym for “Linux, Apache, MySQL, and Perl/PHP/PYTHON”. We are using WAMP, which is an adaptation to a Windows system. MySQL is well adapted to web applications, so it fits in perfect for our project’s needs; he database is called “db_mussel”, Table 3 below show how the data is formatted in this database. The data of each mussel that is of interest is collected here, like which tank the mussel is in, its location at a given time and date, etc. There are two tables in this database “mussel_image_db” and “mussel_location_db” respectively, one for the data collected by image processing pipeline and the other via RFID collection. The structure of the database is shown in Table 4. This information than can be queried through MySQL to study any correlation in behavior to conclude if the backpacks mounted on top of the shell of the mussels affects its natural behavior.

Name of Field	Type	Description
Tank	INT	The tank the mussel was detected in, possible values are in range of 1,2
Mussel_id /color	VARCHAR	Unique ID from FRD tank or color of mussel detected
X_coord	INT	*X*_coordinates of traverse location or centroid information
Y_coord	INT	*Y*_coordinates of traverse location or centroid information
Cycle	INT	Which cycle it was detected in
Time	TIME	Time stamp of when the mussel was detected

Table 3 – Table Showing MySQL Database Field Descriptions.

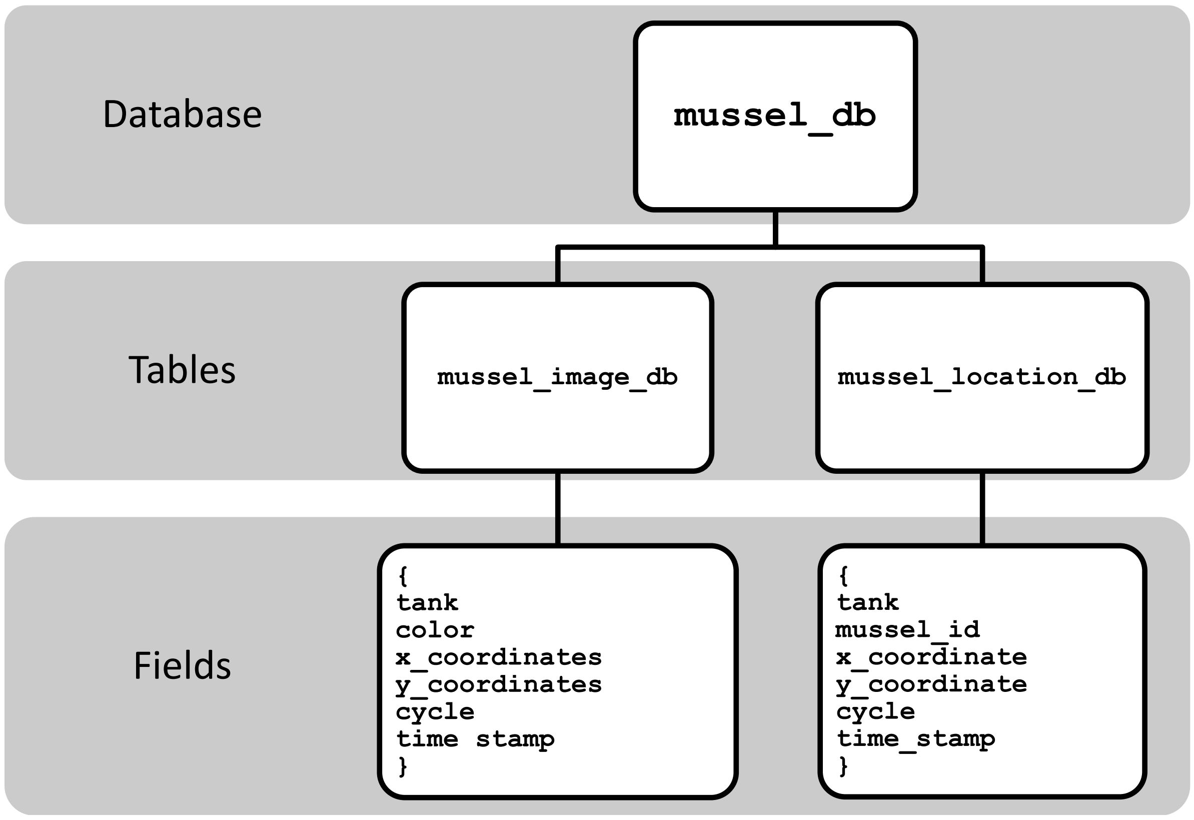

Database
mussel_db
Tables
mussel_image_db
mussel_location_db
Fields
{ tank color x_coordinates y_coordinates cycle time stamp }
{ tank mussel_id x_coordinate y_coordinate cycle time_stamp }

Table 4 – Table Showing MySQL Database Layout.

INTEGRATION

The purpose of this chapter is to describe how all the components of the modular design interact with each other. The main components are Camera Control application; this is the beginning of the whole process, it intakes user data for camera setup and generates JPG Images that are stored in a structured manner in the Hard disk.

- Image Processing; this application intakes JPG Images and generates co-ordinate data that is stored in the MySQL database.
- Video Generation; this application intakes JPG Images and generates videos that are stored in the hard drive.
- RFID-Traverse; has no inputs from other applications, detects mussels via RFID and logs co-ordinate information in the MySQL database.
- Web Server; inputs are Images, MySQL database and Videos and output is a website.

The flowchart below represents this information visually, all the processes are location in the peripheries of the diagram and all the data is located in the center.

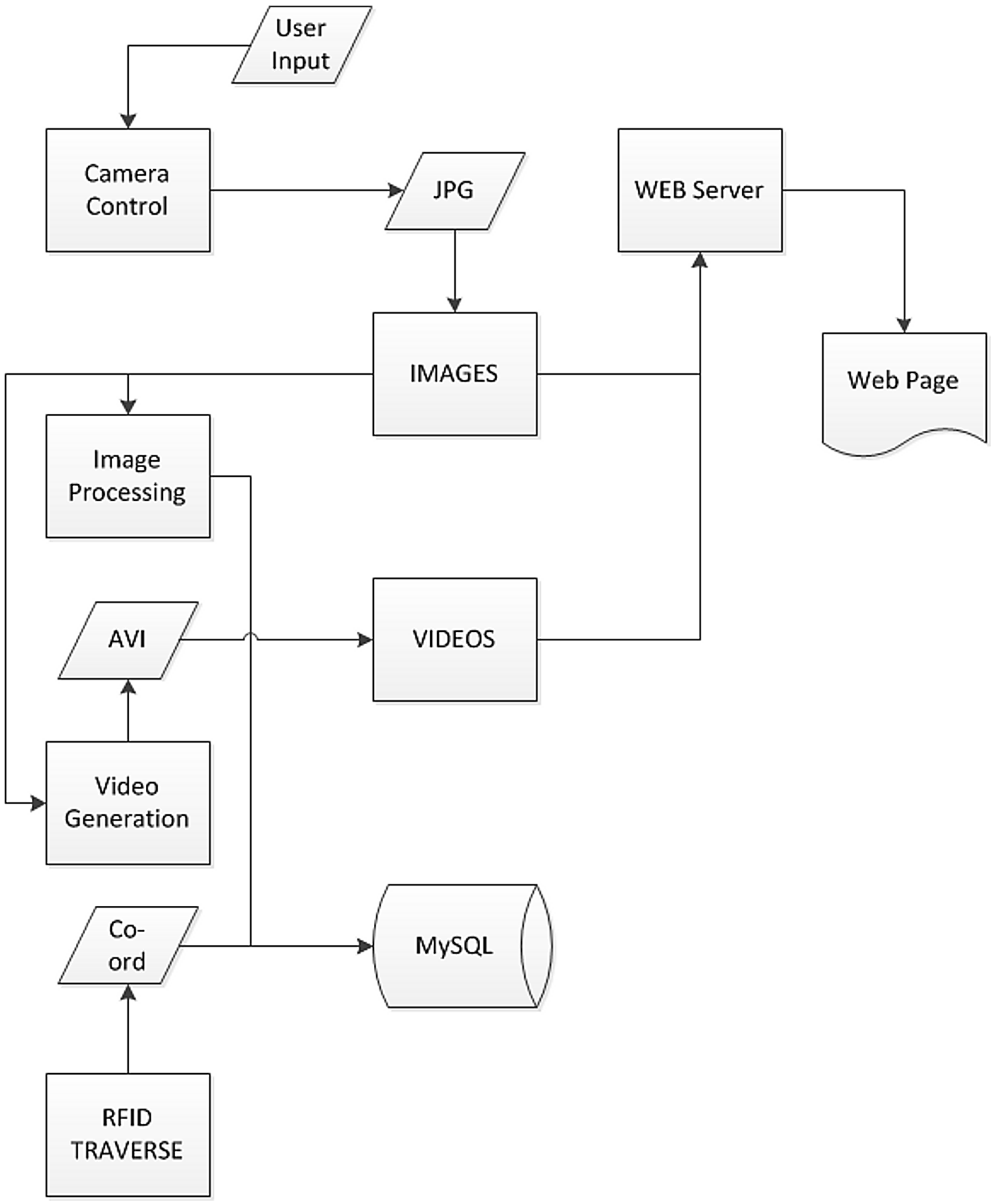


Figure 45 – Flowchart Showing How Data Flows Between Each Module.

CONCLUSION

This project focused on providing a reliable and user-friendly way to track the movement of freshwater mussels in a laboratory environment. This work is part of a larger investigation the behavior of freshwater mussels in a mesohabitat used by the Cybermussels group at the University of Iowa. Tracking mussel movement poses unique challenges. First, the mussels are living organisms, and we need to track their movement in water. They are filter-feeders that get their food from suspended material in the water column. Consequently, imaging techniques must "see through" fairly murky water. Additionally, the mesohabitat simulates a diurnal cycle, so tracking techniques must be able to be effective in both lighted and dark conditions.

We explored the following: time-lapse still-camera tracking, bar-code style identification using fluorescent paint, LED tagging of mussels, and RFID technology.

Time-lapse imaging results were successful. We created a setup with a camera to capture pictures in fixed intervals and make them available to the users through a web interface. The images are used to generate videos, use image processing to detect individual location. Identification through LEDs at night, unique barcodes for labeling each mussel did not yield promising results. However UV lights for illuminating shells with fluorescent paint and RFID tagging for identification worked well in our setup.

REFERENCES

Avago Technologies. (2008). T-1(3mm)Diffused LED Lamps. Retrieved from www.avagotech.com

Bhuptani, M., & Moradpour, S. (2005). RFID Field Guide Deploying Radio Frequency Identification Systems. New Jersey: Sun Microsystems, Inc.

Bourdon, R. (n.d.). WampServer. Retrieved 2011, from WampServer: http://www.wampserver.com/en/

Bradski, G., & Kaehler, A. (2008). Image Parts and Segmentation. In G. Bradski, & A. Kaehler, OpenCV Computer Vision with the OpenCV Library (1st Edition ed., pp. 265-295). Sebastopol: O'Reilly Media, Inc.

Bradski, G., & Kaehler, A. (September 2008). Tracking and Motion. In G. Bradski, & A. Kaehler, OpenCV Computer Vision with the OpenCV Library (1st Edition ed., pp. 337-341). Sebastopol: O'Reilly Media, Inc.

Brandaski, G., & Kaehler, A. (2008). Contours. In G. Brandaski, & A. Kaehler, OpenCV Computer Vision with the OpenCV Library (1st Edition ed., pp. 222-264). Sebastopol: O'Reilly Media, Inc.

Buch, N., Yin, F., Orwell, J., Makris, D., & Velastin, S. (2009). Urban Vehicle Tracking Using a Combined 3D Model Detector and Classifier. In N. Buch, F. Yin, J. Orwell, D. Makris, & S. Velastin, Knowledge-Based and Intelligent Information and Engineering Systems (pp. 169 - 176). Berlin / Heidelberg: Springer .

Canon. (2012). Canon Digital Camera Software Developers Kit. Retrieved 2011, from Canon: http://usa.canon.com/cusa/consumer/standard_display/sdk_homepage

Canon. (2012). Canon Digital Camera Software Developers Kit. Retrieved 2011, from Canon: http://usa.canon.com/cusa/consumer/standard_display/sdk_homepage

Cardona, A. (2011, 10 16). A Fiji Scripting Tutorial. Retrieved 2011, from http://www.ini.uzh.ch/~acardona/fiji-tutorial/

Comaniciu, D., Ramesh, V., & Meer, P. (2003). Kernel-based object tracking . Pattern Analysis and Machine Intelligence, IEEE Transactions on , 564 - 577.

Effects of overdrive on 5 mm LEDs. (2006, 07 26). Retrieved 2011, from Hobbyist LED information Website: http://www.molalla.net/members/leeper/5mmdeg.htm

Ferreira, T., & Rasband, W. (2011). 26.8 Binary. In T. Ferreira, & W. Rasband, ImageJ Users Guide (p. 98). http://rsbweb.nih.gov/.

Ferreira, T., & Rasband, W. (2011). 26.8.8 Fill Holes. In T. Ferreira, & W. Rasband, ImageJ Users Guide (p. 100). http://rsbweb.nih.gov/.

FFMpeg. (2012, February 23). ffmpeg Documentation. Retrieved 2012, from FFMpeg: http://ffmpeg.org/ffmpeg.html

FFmpeg. (2011, February 20). Zeranoe FFmpeg builds. Retrieved from FFmpeg: http://ffmpeg.zeranoe.com/builds/

FIJI. (2012, January 19). Fiji Is Just ImageJ. Retrieved 2011, from http://fiji.sc/wiki/index.php/Fiji

FIJI. (2011, 10 19). Jython Scripting. Retrieved 2011, from http://fiji.sc/wiki/index.php/Jython_Scripting#Jython_tutorials_for_ImageJ

FIJI. (2010, June 7). Nuclei Watershed Separation. Retrieved 2011, from http://fiji.sc/wiki/index.php/Nuclei_Watershed_Separation

Gonzales, R. C., & Woods, R. E. (2002). Color Image Processing. In R. C. Gonzales, & R. E. Woods, Digital Image Processing (2nd Edition ed., pp. 290-302). New Jersey: Prentice Hall.

Gonzales, R. C., & Woods, R. E. (2002). Image Compression. In R. C. Gonzales, & R. E. Woods, Digital Image Processing (2nd Edition ed., pp. 453-454). New Jersey: Prentice Hall.

Gonzales, R. C., & Woods, R. E. (2002). Image Enhancement in the Frequency Domain. In R. C. Gonzales, & R. E. Woods, Digital Image Processing (2nd Edition ed., pp. 175-178). New Jersey: Prentice Hall.

Gonzales, R. C., & Woods, R. E. (2002). Image Enhancement in the Spatial Domain. In R. C. Gonzales, & R. E. Woods, Digital Image Processing (2nd Edition ed., pp. 88-90). New Jersey: Prentice Hall.

Gonzales, R. C., & Woods, R. E. (2002). Image Segmentation. In R. C. Gonzales, & R. E. Woods, Digital Image Processing (2nd Edition ed., pp. 595-611). New Jersey: Prentice Hall.

Gonzalez, R. C., & Woods, R. E. (2002). Image Segmentation. In R. C. Gonzalez, & R. E. Woods, Digital Image Processing (2nd Edition ed., pp. 617-624). New Jersey: Prentice Hall.

Harris, T. (2012, February 21). How Fluorescent Lamps Work. Retrieved from HowStuffWorks.com: http://home.howstuffworks.com/fluorescent-lamp.htm

Hawxby, R. M. (n.d.). Library Electronic Plublishing Center. Retrieved 02 2012, from Oklohoma State University: http://digital.library.okstate.edu/oAs/oas_pdf/v57/p54_60.pdf

Hunt, V. D., Puglia, A., & Puglia, M. (2007). RFID A GUIDE TO RADIO FREQUENCY IDENTIFICATION. New Jersey: John Wiley & Sons, Inc.

ImageJ. (2012). ImageJ. Retrieved 2011, from ImageJ Image Processing and Analysis in Java: http://rsbweb.nih.gov/ij/

Isard, M., & Blake, A. (1996). Computer Vision — ECCV '96. In M. Isard, & A. Blake, Contour tracking by stochastic propagation of conditional density (pp. 343-356). Berlin / Heidelberg: Springer .

Jing Zhou, J. P. (2011, October 14-16). The Specific Moving Target Tracking in the Complex Background. SciVerse ScienceDirect , 806-811.

Kingbright. (2011, June 15). 2.0x1.25 mm SMD CHIP LED LAMP. Retrieved from http://www.kingbrightusa.com/images/catalog/SPEC/AP2012CGCK.pdf

Lahiri, S. (2005). RFID Sourcebook. New Jersey: Pearson Education Inc.

Lowe, D. G. (2004, 11 01). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision , 91-110.

Mancuso, M., & Battiato, S. (2001). An Introduction to the Digital Still Camera Technology. ST Journal of System Research .

MPlayer. (n.d.). Chapter 7. Encoding with MEncoder. Retrieved 2011, from MPlayer - The Movie Player: http://www.mplayerhq.hu/DOCS/HTML/en/mencoder.html

MPlayer. (2011, February 20). Download. Retrieved from The MPlayer Project: http://www.mplayerhq.hu/design7/dload.html

MySQL. (2012). MySQL. Retrieved 2012, from Developer Zone: http://dev.mysql.com/

NAE. (2008). Grand Challenges for Engineering: Managing the Nitrogen Cycle. Retrieved from National Academy of Engineering: www.engineeringchallenges.org/cms/8996/9132.aspx

Nakamura, J. (2006). Shot Noise. In J. Nakamura, Image sensors and signal processing for digital still cameras (pp. 74-77). CRC Press.

Pidwirny, M. (2006). The Nitrogen Cycle. Retrieved 02 27, 2012, from Fundamentals of Physical Geography, 2nd Edition.: http://www.physicalgeography.net/fundamentals/9s.html

pySerial. (2010). Welcome to pySerial's documentation. Retrieved 2011, from PySerial: http://pyserial.sourceforge.net/

Python. (2011). Python. Retrieved from http://python.org/

Rabalais, N., & Turner, R. (2001). Hypoxia in the Northern Gulf of Mexico: Description, Causes, and Change. Coastal hypoxia: consequences for living resources and ecosystems; coastal and estuarine. American Geophysical Union , 1–36.

Richard Y. D. Xu, J. G. (2004). Robust Real-Time Tracking of Non-rigid Objects. VIP '05 Proceedings of the Pan-Sydney area workshop on Visual information processing (pp. 95-98). Darlinghurst: Australian Computer Society, Inc. Darlinghurst, Australia, Australia ©2004.

Science News. (2009, May 27). Light Impacts. Retrieved 2011, from Science News: http://www.sciencenews.org/view/feature/id/7375/title/Light_Impacts

Stevenson, R. (2009, August). The LED's Dark Secret. Retrieved 2011, from IEEE Spectrum: http://spectrum.ieee.org/semiconductors/optoelectronics/the-leds-dark-secret/0

Tao, F. T. (2005, October). Object tracking with dynamic feature graph. Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on , 25 - 32.

Tao, H., Sawhney, H., & Kumar, R. (2002, January). Object tracking with Bayesian estimation of dynamic layer representations. Pattern Analysis and Machine Intelligence, IEEE Transactions on , 75-89.

TROJANUV. (2012). UV RESOURCES. Retrieved February 21, 2012, from TrojanUV: http://trojanuv.com/resources/trojanuv/Backgrounder/UV_Basics.pdf

Turner, R. N. (2008). Gulf of Mexico Hypoxia: Alternate States and a Legacy. Environmental Science & Technology , 2323–2327.

US Department of Energy. (2006). Lifetime of White LEDs. US Department of Energy.

Yilmaz, A. J. (2006). Object Tracking: A Survey. ACM Computing Surveys , Article 13.