Article

An Affordable, Autonomous, Solar Powered and Modular Robotic Water Monitoring System

By Raghav Samavedam, Jayanth Gunda, and Ryan Ziegler

Introduction

Ohio is no stranger to water quality problems, with some of them garnering national and international attention. In 2011, Lake Erie, which borders 4 U.S. states and Ontario, Canada, had a record setting algal bloom that covered 2000 square miles, three times larger than any other previously observed [1], [2]. This phenomena, Cyanobacterial Harmful Algal Bloom (CyanoHAB), primarily occurs due to excessive nutrient presence from agricultural fertilizer and urban runoff, as well as improper wastewater discharge [1]– [3].

In the case of the 2011 bloom, unusually high water temperatures and phosphorus levels - the latter likely caused by fertilizer runoff from northern Ohio farms - combined to nurture and grow the algal population in the lake [4]. When this algal population grows to a critical mass, its toxin release can negatively impact humans and animals drinking this contaminated water by causing skin irritation. Other problems, including respiratory, gastrointestinal and neurological problems, are particularly dangerous to certain high risk groups such as people with chronic diseases, children, and the elderly [5].

Furthermore, algal bloom growth and its subsequent decomposition depletes the water body of dissolved oxygen, resulting in a higher death rate for fish and other aquatic organisms. Though this problem had its roots in northern Ohio, toxic CyanoHABs have been implicated in human illness and death in 43 states nationwide [3].

Local pond health is also a subject of concern regarding water quality in Ohio. In July of 2018, Governor John Kasich signed an executive order declaring eight major watersheds in Ohio as distressed in order to target runoff from agricultural fertilizer applications. Municipal wastewater effluent has also been cited as a threat to local pond health. It has been termed as inadequately treated sludge discharged into the environment. Moreover, this effluent may contain organic emerging contaminants such as pharmaceuticals, illicit drugs and personal care products, which in turn could lead to endocrine disruption in local wildlife. Petrie et. al. [6] notes that these problems are hard to monitor citing that "[d]uring wastewater treatment, there is a lack of suspended particulate matter analysis due to further preparation requirements and a lack of good analytical approaches."

And in the context of private ponds, Pennsylvania State University [7] found that most pond owners have never tested their ponds, and water quality problems are usually only detected after they cause a problem.

Both problems - algal blooms in large lakes and sewage effluent discharge in small water bodies - can be mitigated with effective water monitoring techniques [3], [8], [9]. Real-time information on water quality is very important for predicting major pollution problems in lakes and directing active management of the issue so as to ensure our water resources long-term availability [8].

However, Contemporary water monitoring has its own set of complex issues. Most water quality issues arise from diffuse non-point sources like agricultural runoff from farms and animal feeding lots, the same type of pollution that is widely believed to cause CyanoHABs in Lake Erie. Compared to contaminants that come directly from point sources (e.g. effluent discharge from one sewage treatment center), agricultural runoff is harder to monitor, evaluate, and control primarily due to the fact that the emission of pollutants

from a non-point source is highly variable along short periods of time. With considerable variation even on an hourly basis, single measurements are not able to characterize the water quality and quantify pollution levels over time. Thus, frequent samples in multiple areas are needed for accurate representations and models of water-body pollution.

It is here that we see the drawbacks of current water monitoring systems. Standard water quality testing options are generally heavily reliant on human fieldwork to gather test samples and ex-situ lab testing. Frequent sampling at multiple areas becomes a massive logistical and financial problem due to the human factor involved and the need to send samples to off-site laboratories for further processing introduces a considerable time delay. For these reasons, standard methods fail in helping to monitor, evaluate and model non-point sources of pollution in real time.

Current research has tried to address these shortcomings. Solar-powered buoys have been created that contain water monitoring sensors for temperature and dissolved oxygen among other factors. To combat the fact that buoys are stationary, mobile remotely-controlled boats with similar sensors have also been developed that are able to operate for short periods as opposed to continuous, long-duration deployment. However, these systems are expensive, costing about USD 6,000 - 7,000 for the buoys and USD 30,000 for other stationary, automated water monitoring systems.

In this project, we seek to expand on the work of Dunbabin et. al. from the University of Queensland [10] and create an affordable solar powered autonomous raft for water monitoring of non-point and point sources of pollution. We stress affordability by building our robot with components one can easily find in a hardware store all for under USD 1,500 by using a design that is modular and makes the robot easy to transport and service.

In what is a clean break from past research, we focus on ensuring our water-monitoring system can innovatively communicate its results by building a web client to visualize our collected data and developing a natural language parser interface so that users of all educational levels can intelligently query for specific pollution details.

We also lay out a framework for predictive analysis of our data and future plans of study involving dynamic water treatment.

The following concepts detail the important terminology and framework, in the context of our research, to be used to the experiment:

Design and Development

With water crises in places all over the United States, from the foothills of Appalachia to the urban spaces of Compton, California, there is a need not only for reliable water monitoring but also for an affordable system that cash-strapped municipalities can afford. Thus, in creating this water monitoring system, we have stressed cost-affordability and usability as our mobile water monitoring system costs under USD 1,500 compared to other stationary systems that start around USD 5,000 and require human fieldwork which is economically infeasible for providing enough number of data points over long periods of time.

As water monitoring devices are expensive, we wanted to keep the main design of the autonomous robotic device as modular and inexpensive as possible. For our flotation devices, we used two sealed high density polyethylene pipes as they are biodegradable and relatively inexpensive. Through calculations on the displacement of the flotation devices, we determined that the upper limit on the mass of the robotic device was about 25 pounds or 11.4 kilograms - this meant that we could carry an additional 8 pound (3.6 kilograms) payload that could be used to do additional testing.

This weight load is possible as the raft is formed with lightweight aluminum extrusion rods joined together using L-brackets and stainless steel screws and nuts which are all available at most hardware stores.

For propulsion, the robotic device used two 22 Watt hexagonal-shaft motors, each connected directly via shaft to a custom-designed, 3D printed paddle wheel comparable or cheaper in cost to other alternatives in the market. The motors were controlled by an affordable, multi-interface hardware controller connected to a smartphone, which had GPS capabilities for autonomous navigation and cellular capabilities for data upload and manual control.

(See Figure 1 for Robot Design in Action)

In order to maximize uptime and reduce the need for human intervention to do charging, a 50 Watt solar panel is attached to the top of the robotic raft for powering the onboard electronics and motors. We initially researched a tilt mount to maximize solar output.

However, we ultimately decided against this due to stability concerns and the need for the robotic device to be oriented such that the panel faced true south at all times. For regulating and storing energy created by the panel, the robot device uses a charge controller and 200-watt hour lithium- phosphate battery. Although the parts that form the solar power system (panel, charge controller and battery) are the most expensive components of the robotic raft, they add critical functionality for the robot to have long-term use (See Solar Panel, Power and Battery Considerations for more information).

Existing water monitoring systems derive most of their cost from their sensors and sample testing, so in our goal to develop an affordable water monitoring solution we focused on adding water sensors that could give good all-around insight into the environment but still be inexpensive cost-wise. In its current configuration, the autonomous robotic device uses a modified fish tank sensor that is able to detect ammonia, pH, water temperature and light intensity (See Systems Applications to Environmental Problems for further environmental insight said parameters can give). This sensor is able to measure free ammonia through detecting the degree of color change in a specially designed slide that is periodically exposed to the surrounding water. The readings are reliably accurate and though the slide needs to be replaced every month or so, it is relatively inexpensive - costing about 10 USD at the time.
The readings from the sensor are extracted and uploaded by a Raspberry Pi micro-computer (See Data Management and Access for more details) which can theoretically interface with any sensor that has a USB connector, making the robotic device modular and allowing for future sensors to be added to analyze the lake (See Figure 2).

Autonomous Navigation

The system is equipped with two Motorola G5 Plus Android phones: One on the robotic raft, and one with the client operator. These phones are connected through single radio hop Wi-Fi Direct communication, which allows wireless connection to be established without the need of a wireless access point. This dependency allows the usage of Global Positioning System Location Tracking on the onboard phone to obtain the Latitude and Longitude coordinates of the robot position. The compass sensor capability of Android is also employed by using the pre-built phone magnetometer to return compass readings that determine orientation.
The autonomous navigation algorithm uses the dimensions of a given water body to generate waypoints that provide a maximum traversal of the area. This provides sufficient sampling variability to eventually yield the most complete picture of the water quality that can be used for further analysis. The waypoints are placed in an order relative to the initial robot position, and their positions are used to divide the lake into various attention sub-regions (Fernandes et al) [11]. Next, a minimum length tour i.e., the shortest overall path that allows the robot to cover each region, is constructed (Branson et al) [12]. A segmented sequence is created with the ends of each segment corresponding to the relative waypoint order. For each segment, its length and its angle measurement (assuming the standard Cartesian coordinate plane) are calculated. The algorithm then makes use of the encoder capability of the raft motors, which generate a specified number of electrical signals for every rotation of the paddle wheel. A one-to-one correspondence is thus established between the distance and the encoder ratio to determine the number of paddle revolutions, and its corresponding encoder value needed to reach the waypoint destination. This allows the robot to move freely and with minimal interference.
Proportional-Integral-Derivative Control, a control-loop feedback mechanism system, is also used to tune an optimal motor speed consistent with the encoder readings (Pereira) [13]. This creates a navigation system independent of battery life to generate the maximum positional accuracy. Finally, the prior calculated angle is transformed into a yaw rotational value that the robot needs to obtain to be able to move exactly along the line segment. During travel, the GPS coordinates of the current robot position are compared with the end waypoint to act as a calibration check system (Burgard et al) [14]. Based on the real-time encoder count value, the true GPS position that preserves the correction direction (along the line segment) and orientation (same as initial angle) is calculated. This is then compared to the actual data values, and the robot is made to move accordingly until the margin of error reaches an acceptable tolerance level.
(See Figure 4 for Generated Path Example)

Implementation of PD Course Correction Control

To ensure that our robot, in its autonomous operation, stays on course as it moves from target to target, we have implemented a form of PD (Proportional-Integral-Derivative) Control System.
First, before navigation truly begins, the robot, using a GPS Location Service, is able to extract the latitude and longitude of its starting position (known as initial position in the data section). Then, a timer is started to eventually calculate velocity (detailed below). Next, based on the latitude and longitude of its target position, a straight line path (hypotenuse) is drawn from the initial to target, and the corresponding azimuth angle (0 Degrees being North, and rotating counter-clockwise) is calculated based on the trigonometry of the right angled triangle. The robot takes its own azimuth (obtained through the magnetic sensor of the on-board Android Phone), and rotates until it reaches the desired azimuth of the path, plus or minus a threshold of 5 degrees (PI/36 Radians).
Next, the robot sets both of its motors to a power of 0.25, and continues straight along the specified path. This is where the PD Course Correction comes into play. We track two locations: Previous Location which is the latitude and longitude of the robot before the GPS gives an update, and Current Location: The most accurate location of the robot after the update from the GPS Location service. Initially, Previous Location and Current Location are the same, since the GPS Measurements have not changed.
Once the location gets updated by the GPS, Previous Location is no longer equal to Current Location. At this point, we calculate two things for our PD Algorithm. First, we calculate the length of the perpendicular distance from the robot's most updated position to the straight line hypotenuse connecting the initial position to the final position (initially calculated from the first rotation). This is accomplished using trigonometry and vector projections, as shown in Figure 8. Also, as soon as the location is updated, the timer that we set in the original rotation is stopped, and the velocity vector of the robot motion is calculated using the displacement vector and the elapsed time. Using the same principles of trigonometry as before, the orthogonal velocity (velocity in the direction of the original path) is also calculated.
These two metrics combined tell the robot how much to rotate, and in what direction, to get back on track to the original created path. The distance and orthogonal velocity values calculated are linearly combined with P and D coefficients (tuned by Experimental Testing) to give a total angle of rotation: Total = P * Distance + D * Orthogonal Velocity (P: Proportional to Error, which in this case is the distance to the original path, D: Derivative: Rate of Change of Error, which in this case is how fast the robot is deviating away from the path, i.e., the velocity). Roughly, the Distance will tell us how much of a numerical angle to turn, and the Velocity's Direction/Magnitude will help us to control overshoots/undershoots in this rotation.
After this control, the robot gets back on course, and continues in the straight line path again, until the next time the GPS Updates. At that point, the PD will take effect again for course correction. This process will continue until the final target has been reached.
What follows is a tabulation and graphical representation of the autonomous navigation algorithm ()., First, the GPS location of each waypoint is recorded, and for each successive destination pair, the needed angle rotation, and length of travel path are also calculated. (See Table 1)
Next, the periodic check value is documented and compared it to the actual robot position.
(See Table 2)
Finally, the appropriate numerical values are graphically represented on the coordinate axes, which are superimposed onto the region of detection.
(See Figure 4)

Computer Vision

To ensure the avoidance of all obstacles during robot motion, autonomous navigation is supplemented with a Computer Vision Algorithm, which runs using the Open Source Computer Vision Library. The development of the algorithm followed two main steps: Image and Real-Time Video Detection.
The first focus was on tracking various types of static images to help us to identify the most pertinent features that are needed for a comprehensive object understanding. Images chosen were applicable to the robot's scenario, and included boats, wild animals, bunched seaweed, etc. Implementing the Mat Container feature of OpenCV, allowed for obtaining access to the raw image pixelation data, and used the resulting

numerical values as a precedence order for detail prioritization (Baksheev et al) [15]. This led to a narrowing down on four basic key factors: Degree Presence of each of the 7 main colors (White, Red, Green, Blue, Yellow, Brown, and Black, plus any combined shades), concentration of the edge pixels on the object boundary, length of the substance (estimated using the spanned distance of the best geometric figure approximation), and the standard deviation error of the generated shape to the actual physical quantity in the horizontal and vertical directions (Bloisi et al) [16].
Next, the process of detection itself was tuned to apply to a traversal of a lake. Standardizing the images using RGB-Grayscale conversion ensured that the system was illumination-independent (Emami) [17]. As such, the issue of limitation of sample collection to daylight hours was bypassed.
In addition, the software also included the ability to approximate the edges of imperfect instances of objects. This is accomplished through the Hough Transform, which uses a parametric coordinate system to output a best-fit edge curve based on read physical properties (Barngrover) [18]. Through this overestimation, a tolerance level is generated for the raft navigation that provides ample room for maneuverability. After laying out this preparatory groundwork, the process transitions to video. From the OpenCV GPU module, the Android mobile device on the robot becomes capable of stabilizing streaming video and breaking the reception down into an individual frame-by-frame sequence. The same metrics from the images are employed in the video analysis by essentially taking each frame as its own separate quantity. The phone is positioned per a landscape orientation and installed at the front of the robot to maximize the viewing range of its inbuilt camera. Whenever a pressing obstacle, in terms of size and threat level, is detected, the robot stops and begins a counterclockwise rotation until the algorithm deems its orientation as safe to proceed.
In the future, we plan to use techniques from Mathematical Morphology and dilate specific goal pixels to continue to enhance accuracy (Lu and Xie) [19].
(See Figure 7)

Solar Panel, Power and Battery Considerations

Our robotic device uses a 200 Watt Hour Lithium Phosphate battery for storing energy from the 50-watt solar panel, as this battery is significantly lighter than lead acid batteries, has a high number of charge cycles, is non-toxic unlike Lithium-ion or lead acid batteries and less flammable than Lithium-ion batteries. With an estimated 5,000 charge cycles the battery would last about 10 years, making it a durable, sustainable and environmentally friendly option.
Based off of conservative estimates, we calculated that the robotic device would have an instantaneous 50 Watt power draw when moving. So, in choosing battery sizing, we computed that a 200 Watt Hour battery would give enough energy for the robot to move intermittently through the day.

Based on size and weight limitations, a 50 Watt solar panel was integrated into the design with an assumed 6 hours of sunlight a day during monitorable months. Because the robotic device has low amperage draw, the solar panel is connected to the battery via a 30 ampere Pulse Width Modulation (PWM) charge controller to ensure optimum performance at lower cost. The PWM charge controller can power a USB device and the battery, giving added flexibility. Since all the loads are DC powered and can be directly powered by the battery, there is no need for an inverter.
The charge controller can also be fitted with a Bluetooth module in the future to monitor the energy output from the solar panel for educational purposes.

Data Management and Access

In order to effectively collect and act on data from the autonomous robotic device, data must be exchanged between it and the servers. The autonomous robotic device is equipped with a Raspberry Pi, a small computer running a Linux-based operating system. The Raspberry Pi contains a variety of General Purpose Input Output pins, four USB ports, and an RJ45 port to interface with the USB-connected water sensor. The script utilizes the Message Queue Telemetry Transport (MQTT) protocol to send data to the server at frequent intervals. MQTT is a lightweight transport protocol for Internet of Things (IoT) devices to stream data to a server. RabbitMQ converts MQTT messages into Advanced Message Queueing Protocol (AMQP) messages. A service written using JavaScript reads messages from the alert queue, and determines if they match alert thresholds for pollution, as defined by the users. When

a threshold is met, the alert service sends an email to the associated email account. Another service, also written in JavaScript, reads messages from the processing queue and writes them to a database. In addition, as it is essential for researchers and the general public to easily access water pollution data, we also developed an innovative web client interface for communicating collected data. To visualize the data, a heat map displays pollution statistics made up of pH, temperature, and ammonia levels. The user can utilize a slider to see historical pollution levels and trends.
(See Figure 2)

By using multiple services, alerts can be sent at the same time data is being saved, removing a bottleneck present when several devices report statistics at once.
(See Figure 3)
Additionally, users can obtain point data using a Natural Language Processing (NLP) interface. The user can type in a question about specific pollution factors, and the NLP system will process the request and find relevant data for the user. The language processing system extracts keywords, for example temperature, and then analyzes temperature data for the date(s) provided to return data to the user. If a user does not provide a date or time, the parser can extract keywords that indicate tense and relative date (what will be, what was) to determine the date(s) to query for. And lastly, if a user does not indicate a specific pollution factor in their query and instead asks about general water quality, the parser will return a water quality score determined by internal calculations on the central tendency of the pollution metrics being measured.
(See Figure 5)
Both the heat map and NLP interface use a Representational State Transfer (REST) interface to communicate with and obtain data from the servers. Gaining composite data is useful, but is no substitute for raw data, especially for research applications. In the near future, researchers will be able visit a dedicated research portal and request to download data. The portal uses the same REST interface as the heat map to retrieve data. Researchers will simply check boxes to select which data to access, and upon submission of a request, a retrieval service will email researches data. By using a standardized REST interface, researches can also directly query for data instead of using a portal, opening further expansion opportunities.

With these interfaces, our water monitoring system can also have applications in environmental education. Students can learn about water pollution, and apply water pollution concepts to our web application to complete projects. For example, students could use heat maps combined with statistics extracted via NLP to explain algal blooms and detect areas with potential sewage contamination.
Ensuring reliable access to data is a requirement for significant research to be performed. As such, we have taken a multi-cloud approach to reliability. Microsoft Azure is our primary cloud provider. Microsoft Azure is an automatable, secure, and reliable cloud platform with data centers across the United States retrieving services that run on a Kubernetes cluster. Kubernetes is a server and cluster orchestration tool which enables us to easily deploy and scale applications across multiple servers with auto-scaling rules. The cluster will automatically run the number of services required to meet current demand. However, in the event of failure in Microsoft Azure, we have an identical Kubernetes cluster running on Amazon Web Services (AWS). Devices will automatically send data to AWS if Azure is unreachable, ensuring valuable data is not lost.

Research Methods

The methods conducted for our two separate experiments are detailed as follows:

Computer Vision

i) Accuracy
To test the accuracy of the Open-CV algorithm (specifics of which are detailed in the introduction), ten images (See images 8 to 18 in the Figures document) were selected that pertain to possible scenarios that our robot could encounter during its time in the body of water, such as rocks, debris, animals, etc. Each image was placed a constant distance of 8 inches in front of the onboard robot camera. For each trial, the maximum size of the dimensions of the pixel rectangle created by the detection system was recorded. Variations within the generated sizes were noted, as well as any failures/irregularities in the detection. Note that, sometimes, the goal is for the robot to fail in the rec-

ognition of a specified object. Consider the example of lily-pads. In a given lake, there could be numerous lily-pads, ranging from the single to double and even triple digits. It is not worthwhile for the robot to detect and turn away from these plants at every possible opportunity, as it will reduce efficiency in motion. The best option is for the robot to continue along its path since contact with lily-pads will not harm its mechanical structure. Our selected images also encompass this aspect of detection. If there is a huge deviation for any of the image results from trial to trial, a brief explanation is written to explain why this might have been the case.

ii) Orientation
To test the ability of the robot to modify its orientation, the same ten images used in the accuracy test (See images 8 to 18 in the Figures document) are placed in front of the onboard robot camera (through the Moto g5 Plus Android Phone), at a constant 8 inches away. For each image, three metrics are recorded as part of our data. First, a stopwatch is set as soon as the Computer Vision Detection Program is initialized, and the time taken for the phone to output a reading detailing the object (either the dimensions of a pixel size rectangle or a failure to recognize) is recorded. Next the motion of the robot based on the initial detection in step 1 is noted. After each detection, the robot can enter two possible states: It can either go from two wheels spinning to only one, which implies that it is has deemed the detected object as a harm and is turning away from it, or it retains the state of two wheels spinning, which implies that it is going along its current path after having deemed the object a non-hazard to motion. Finally, after this second step, it is documented whether or not the robot action was appropriate given the circumstances presented by the image. For example, if the image was of rocks in the water, the desired motion is to turn to ensure that there are no crashes. However, if the image was of shrubs in water, there is no need for the robot to rotate, and it can continue going forward on its straight path. All of these values are recorded, and brief explanations are given for any irregularities.

PD Course Correction Control

The intent of the PD Algorithm is for the robot to accurately move from an initial location to a specified target location, which is dependent on the concentration of pollutants in the water. To do this, the robot first starts in an initial position, rotates until it reaches the angle corresponding to the target position (Based on the azimuth reading of the Moto G5 Magnetometer sensor Android Phone). Then, it continues along this straight path. Every time the location updates on the GPS, a 'Turn by PD' function is executed to ensure that the robot re-orients itself to align on the initially calculated path. In this experiment, it is sought to test the accuracy of this function.
A land-based test was chosen for this experiment as it allows more control in robot positioning and data collection. Three initial locations were chosen, with a constant final location for each trial. For each location, we ran exactly one iteration of the PD Course Correction, occurring when the GPS first updated on the Android Phone. During the PD Turn, the change in the azimuth heading of the robot (that is, its orientation) is recorded, as well as the length of the perpendicular vector that separates the initial correct trajectory of the robot (calculated right after the autonomous program is initialized) and the robot's deviated trajectory obtained in the time intervals between location updates. The angle of separation between these two directions is also calculated and displayed in our data table. Finally, after the PD modification has run its course, the total displacement of the robot, from the initial position to the location that the robot's most current position just after the PD Control, is calculated and expressed as a matrix containing both the latitude and longitude components. And, this updated location is also placed in the table as a latitude and longitude pairing. See Figure 8 for a graphical visualization of this course correction experiment.
The objective of this experiment was to be able to numerically quantify all of the values associated with our PD Control (See diagram in the images section regarding PD vectors/angles). And, based on the data gathered, a conclusion can be made as to whether or not our algorithm was accurate enough to ensure constant travel along the correctly calculated path (See results/conclusion section).

Results

See the separate figures document for specific data values tabulated over the course of the experiment. (Specifically, Tables 3, 4, and 5)

Computer Vision Accuracy

Overall, we see that our OpenCV detection software was accurate 44 times out of 50 total trials (5 attempts for each image, with 10 images in total). This is an accuracy rate of 88 percent, and is ideal for the testing stage of our robot development. It is also almost ready for on-field usage in the water. The few discrepancies in recognition were either caused by over-saturation of color in the RGB to Grayscale conversion, or distortion caused by the reflection of sunlight rays onto the water, which modified the coloration values as perceived by the vision. To rectify this, the coloration algorithm needs to be modified slightly to account for the cases where extreme shades of color are present. In addition, the orientation of the camera needs to be angled slightly upward so that there is no longer interference by the waves in the water in influencing the robot's decision making.

Computer Vision Orientation

For the ten images put to test the robot's decision making ability, it performed the appropriate action (i.e., turning away for the object or going along the intended path) a total of 9 times out of 10. This is an accuracy rate of 90 percent, slightly better than the accuracy test, and still ideal for the testing stage of our robot development. The one error in motion came from the robot on-board camera focusing on another prominent color, that of the vibrant water waves, as opposed to the sharks, fish, and whales that were somewhat blending in with the water. As such, the CV only identified the presence of water and failed to recognize the aquatic life altogether. To rectify this in the future, we need to modify the algorithm in the cases where the object is fully or partially submerged in the water, so that the coloration process can account for the effect of the color of the water somewhat masking the color of the objects.

PD Course Control

It can clearly be seen that, in all three cases of the angle modification, the azimuth heading was correctly changed to a value that aligned with the initial created path, with minimal error. No matter how much deviation was done on the robot's path through manual displacement and rotation, the angle change was correctly able to be made, allowing the robot to continue on the correct path to reach the target position. In the first trial, it was noticed that the tested P and D coefficients (initially P = 1000, D = 1000) had room for some improvement. While the robot was able to move to the general direction of motion through the course correction, since the two coefficients were quite large, the robot would quickly rotate to be within the threshold specified in our code and not slow down enough as it neared the required azimuth, leading to an accurate, but not precise, turn. To rectify this minor issue, in the second and third trials, the P and D values were lowered, finally settling on P = 891, and D = 928. This gave us, based on the visual observation, the most accurate motion corrections.

Conclusion and suggestions for future research

Conclusion

The conclusion that can be made from this experiment is that our robot is a viable alternative to existing water monitoring systems, in terms of affordability, autonomous control and ease of navigation, and accuracy and precision in data control. Going forward, the aim is to expand upon our system and make it fully robust in all three stages of the process: autonomous navigation, data collection, and server transmission.

System's Applications to Environmental Problems

a) Point Sources

This system has potential for monitoring and evaluating point sources of pollution through its many on-board sensors. Current detection systems are stationary and therefore require an expensive group of sensors to obtain a metric for the entire lake quality from just one sample region. Our robot uses only one main sensor package with the ability to detect multiple contami-

nants at once. Due to its mobility, there is no need to use a network of many sensors as our robotic device is able to relocate this package to many points of a water body on time-scheduled intervals. This results in ease of deployment and allows regulatory agencies to make impromptu, discreet checks on suspects of point-source pollution Moreover, the water quality of the entire body is determined at a much more affordable rate.
Since our robotic device contains a pH sensor, it is possible to detect pollution from improperly managed mining areas by testing a nearby water body for unusually low pH (acidic) levels. At the other extreme, high pH levels can indicate that there is wastewater contamination that contains detergents, soaps and other toxic chemicals.
In areas with negligible agricultural and municipal runoff, the presence of ammonia and other nitrogenous compounds is also a strong indicator of sewage contamination as sewage treatment centers do not remove urea, which later decomposes into ammonia, from wastewater.
This added ability to do wastewater detection gives our water monitoring system an extra dimension in early warning and detection of cyber-terrorism as water treatment centers have been listed as prime targets of cyber-terrorism by the United States Environmental Protection Agency due to the fact that an unauthorized discharge of untreated sludge could cause catastrophic change to our water resources.

b) Non-Point Sources

As mentioned before, non-point sources have variation in their emission of pollutants over time which makes it relatively hard to quantify and evaluate their impact on the surrounding environment with current methods involving one-time monitoring. Non-point sources are also diffuse and non-homogeneous, making standard methodologies that use a network of floating sensor stations economically impractical over large areas.
The water monitoring system put forth is able to solve these problems of infrequent sampling on large test areas as it contains only one continuously reporting sensor package that can migrate around to multiple user-specified testing points in a water body. Thus, there is less overhead due to the fact that there is no need to set up multiple different testing sites and there is added ability to do continuous water monitoring over long periods of time, a trait considered critical for future water monitoring systems. This additional flexibility to quantify non-point sources allows our water monitoring system to tackle pertinent problems in water quality.
Problems in the Great Lakes such as CyanoHABs, which are nurtured by warm water and nutrients from agricultural runoff, can be effectively modelled through use of the onboard temperature and ammonia sensors (refer to concepts of affordability). Ammonia is especially useful in detecting nutrient runoff as most nitrogen-based fertilizers either contain ammonia or turn into ammonia through the nitrogen cycle and most phosphate-based fertilizers consist of di-ammonium phosphate which breaks down into phosphate and ammonia after being released.
Not only as a cause for CyanoHABs, non-point pollution such as nutrient runoff is also a dominant problem for local watersheds [20]. The United States Environmental Protection Agency (USEPA) has for the past several decades' ranked nutrient pollution as one of the top threats to U.S. streams and the United States Geographical Survey (USGS) has prioritized comprehensive national studies to assess nitrogen and phosphorus content in local streams and groundwater [21].
With major focus on national-scale water issues such as Great Lakes algal blooms, important local water issues may be overlooked. As our water monitoring system is affordable, costing under USD 1,500 and requiring no technical background to deploy, local municipalities may find this a useful technology for monitoring diffuse and potentially harmful runoff into their water bodies and sources.

c) Public Health and Early Warning System Potential

The data our robotic device collects is suitable for determining public health considerations as the pol- lution factors the robotic devices sensor can detect - pH, ammonia and water temperature - are useful in gauging environmental health. Any extrema in their values would indicate dangerous consequences for human and wildlife health with at-risk groups such as children, pregnant women, people with chronic disease and the elderly being especially susceptible.
pH is important as fish and other marine biota have a very specific pH tolerance - low pH inhibits shell growth, fertility and at very low levels can cause fish

death. Similarly, on the other side of the scale, higher pH (i.e. basic) water can result in skin and gill issues and fish death in the presence of ammonia.
Ammonia, as well as other forms of nitrogen, in high levels can cause harm to the environment by acidifying the soil and decreasing biodiversity. Ammonia is also dangerous to pregnant women and children and with prolonged exposure can cause burning of the eyes, ears, throat, and lungs.
Extreme values in temperature are detrimental to fish as certain species have very specific tolerance levels. In general, warmer water from runoff reduces dissolved oxygen in the water body which results in reduced growth, excessive respiration rates, and lower survival rates in fish population. For communities that rely on fishing as a source of recreation, income, or both, extreme water temperatures are thus problematic.
And with crises like the Flint Water Crisis garnering national attention, there is clearly a need not only for public health testing of the pollution parameters described above, but also for reliable early warning systems. For any early warning system to work, continuous monitoring of the local water bodies must be done. The water monitoring system we have put forth has this capability due to the autonomous robotic device being solar powered, resulting in minimal downtime for charging or refueling.
Communication of data is also a critical feature of early warning systems. This is where the water monitoring systems web client is particularly useful, as its natural language parser and interactive heat maps are useful not only for environmental education but also for members of the general public who want intuitive, up-to-date information about local water quality.
By combining the current web client with text-message warnings from the server, our water monitoring system would be able to communicate real time water quality data to the public in an innovative way.

Future Research Plans

Once enough data has been collected, we plan to do time series analysis on our collected data to identify three predictive metrics for each collected pollution factor: long term trends, daily forecasts, and seasonal patterns. For the long term-trends, we would employ a linear regression model through a Least Squares Regression Line to maximize accuracy. For both the daily forecasts and seasonal patterns, we would use a Fourier Series to represent the data. The Fast Fourier Transform would allow us to extract the sinusoidal waveforms of our measured metrics. Taking the high-frequency signals, we could determine short-term, daily patterns. Similarly, we would use the low-frequency signals for seasonal to yearly changes. Furthermore, we plan to extrapolate any trends we would discover from our model and use them to predict future conditions, adding extra functionality to an early warning system.
We also have plans to expand our water monitoring system to include dynamic water treatment through automated floating beds of beneficial native Ohio wetland plant species. These species are known for their ability to absorb excessive nutrients such as phosphates, nitrates and iron from a water body. A researcher from Ontario, Canada, has developed floating beds which contain rows of cattails with their roots exposed to the water [22]. He has had success cleaning up certain water bodies by leaving these beds out at predetermined locations for months at a time and then cutting off any growth in the cattail plants afterward. We plan to expand on this idea by using beds of beneficial native Ohio wetland plants to clean water and making it into a connected network of solar powered, autonomous devices (similar in function to that of our current robotic device) that move to optimum positions in a water body based off of analysis done by data collected by our current water monitoring system.

Acknowledgments

We would like to thank the Columbus Green Building Forum (CGBF) for all of their help & expertise and thank CGBF's project sponsor - the Ohio Environmental Education Fund (OEEF), for providing USD 1,200 in material costs to make this idea of an affordable, autonomous water-monitoring system come to life. The three authors volunteered their time to execute this project by providing pro-bono services to CGBF to further its educational mission.

We would also like to thank our mentor Anne Fuller from Dublin Schools for all of her valuable advice in project development, both in software development and mechanical design.

References

[1] C. Portis-Hurlbert, “Algal blooms at lake erie,” Kent State University Digital Commons, 2016.

[2] A. M. M. et. al, “Record-setting algal bloom in lake erie caused by agricultural and meteorological trends consistent with expected future conditions,” Proceedings of the National Academy of Sciences, 2013.

[3] USGS, The science of harmful algal blooms, https://www.usgs.gov/news/science-harmful-algae- blooms, Accessed on 2018-11-18, Oct. 2016.

[4] N. I. of Environmental Health Sciences, Harmful algal blooms, https://www.niehs.nih.gov/health/ topics/agents/algal-blooms/index.cfm, Accessed on 2018-11-18, 2018.

[5] E. B. et. al., “Marine harmful algal blooms, human health and well-being: Challenges and opportunities in the 21st century,” Journal of the Marine Biological Association of the United Kingdom, 2016.

[6] B. Petrie, R. Barden, and B. Kasprzyk-Hordern, “A review on emerging contaminants in wastewa- ters and the environment: Current knowledge, understudied areas and recommendations for future monitoring,” Water Research, vol. 72, pp. 3–27, 2014.

[7] P. Extension, Pond ecology, https://extension.psu.edu/pond-ecology, Accessed on 2018-11-20, 2013.

[8] J. L. Graham, N. M. Dubrovsky, and S. M. Eberts, “Cyanobacterial harmful algal blooms and u.s. geological survey science capabilities,” U.S. Geological Survey Open-File Report, 2016.

[9] D. B. Baker, “Regional water quality impacts of intensive row-crop agriculture: A lake erie basin case study,” Journal of Soil and Water Conservation, vol. 40, no. 1, pp. 125–132, 1985.

[10] M. Dunbabin and P. Smith, “High fidelity autonomous surface vehicle simulator for the maritime robotx challenge,” IEEE Journal of Oceanic Engineering, 2018.

[11] L. C. Fernandes, J. R. Souza, P. Y. Shinzato, and G. Pessin, “Intelligent robotic car for autonomous navigation: Platform and system architecture,” Proceedings - 2012 2nd Brazilian Conference on Critical Embedded Systems, CBSEC 2012, 2012.

[12] P. Tokekar, E. Branson, J. V. Hook, and V. Isler, “Coverage and active localization for monitoring invasive fish with an autonomous boat,” International Symposium on Experimental Robotics, 2013.

[13] A. A. de Menezes Pereira, “Navigation and guidance of an autonomous surface vehicle,” PhD thesis, University of Southern California, May 2007.

[14] R. Kummerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, “Autonomous robot navigation in highly populated pedestrian zones,” Journal of Field Robotics, vol. 32, 2014.

[15] K. Pulli, A. Baksheev, K. Kornyakov, and V. Eruhimov, “Real-time computer vision with OpenCV,” Communications of the ACM, vol. 55, no. 6, pp. 61–69, 2012.

[16] D. D. Bloisi, L. Iocchi, A. Pennisi, and L. Tombolini, “ARGOS-venice boat classification,” in 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, Aug. 2015. doi: 10.1109/avss.2015.7301727. [Online]. Available: https://doi.org/10.1109/avss.2015.7301727.

[17] S. Emami and V. Petrut, “Facial recognition using OpenCV,” Journal of Mobile, Embedded and Dis- tributed Systems, vol. 4, pp. 38–43, 2012.

[18] C. M. Barngrover, “Computer vision techniques for underwater navigation,” Master’s thesis, University of California, San Diego, 2010.

[19] G. Xie and W. Lu, “Image edge detection based on OpenCV,” International Journal of Electronics and Electrical Engineering, vol. 1, no. 2, pp. 104–106, 2013.

[20] W.Grayman, R. A. Deininger, and R.M. Males, Design of Early Warning and Predictive Source Monitoring Systems. Awwa Research Foundation and American Water Works Association, 2001.

[21] S. Ahuja, Monitoring Water Quality: Pollution Assessment, Analysis, and Remediation. Elsevier B.V., 2013.

[22] M. Curry, “Floating Water Filters,” The Western Producer, Glacier Farm Media, 2017.

Figure 1: Autonomous Robot in action in the lake

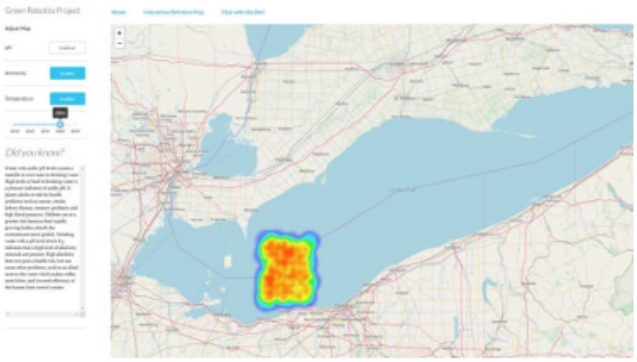

Figure 2: Visual Heat Map created by gathered data

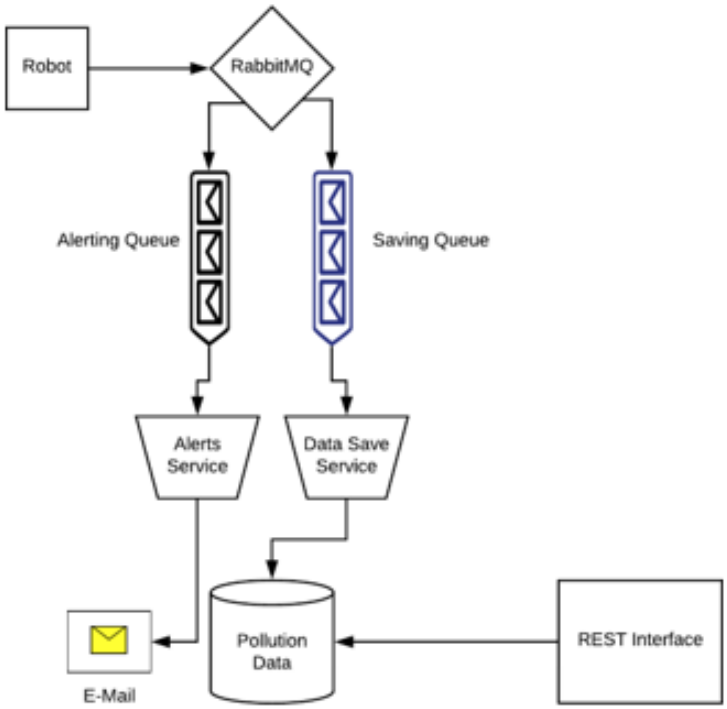

Figure 3: Data Management and Access System

Figure 4: Path traversal generated by Autonomous Algorithm

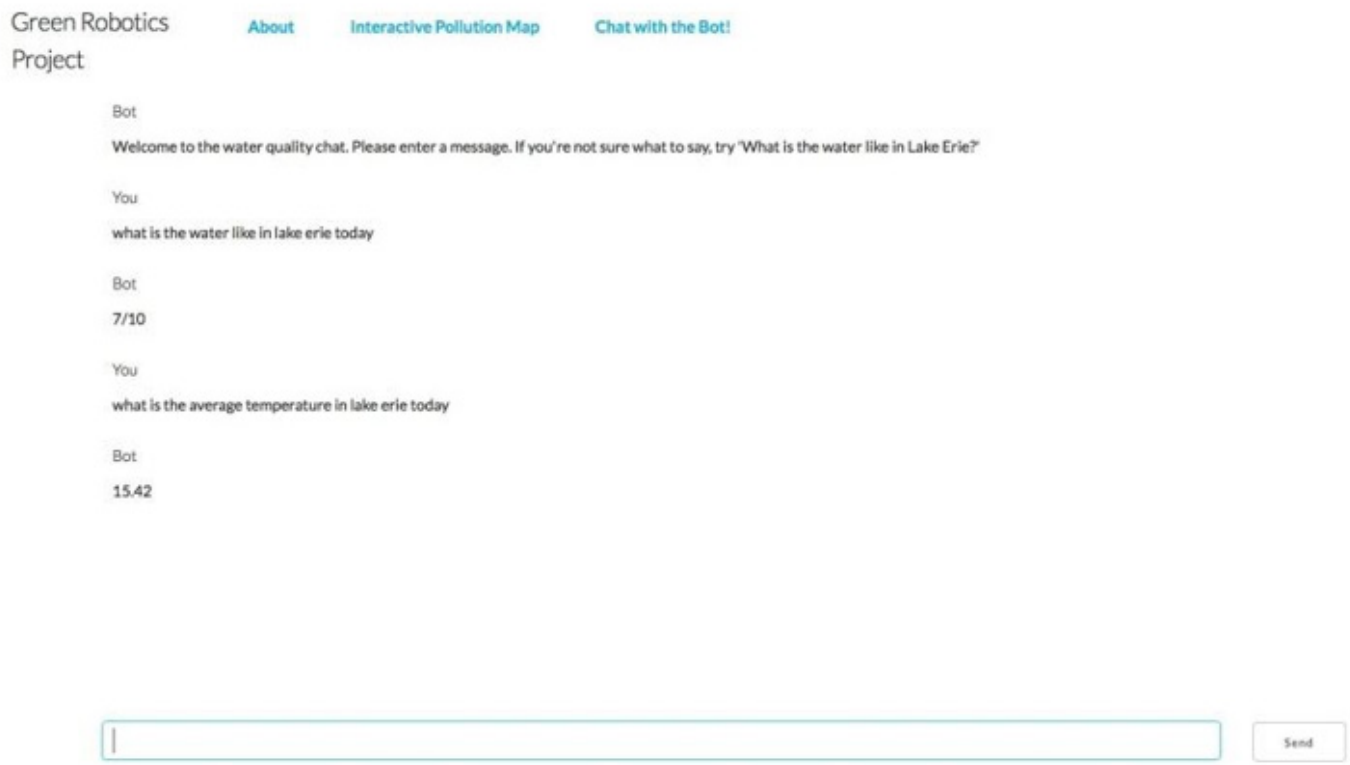

Figure 5: Natural Language Processing Interface

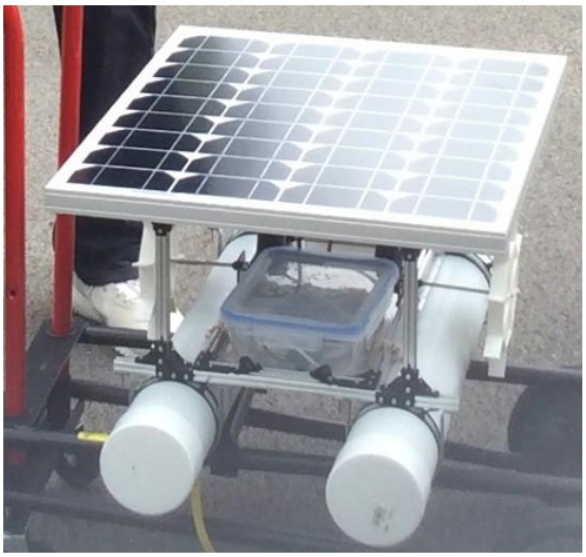

Figure 6: Prototype of the lake monitoring robot

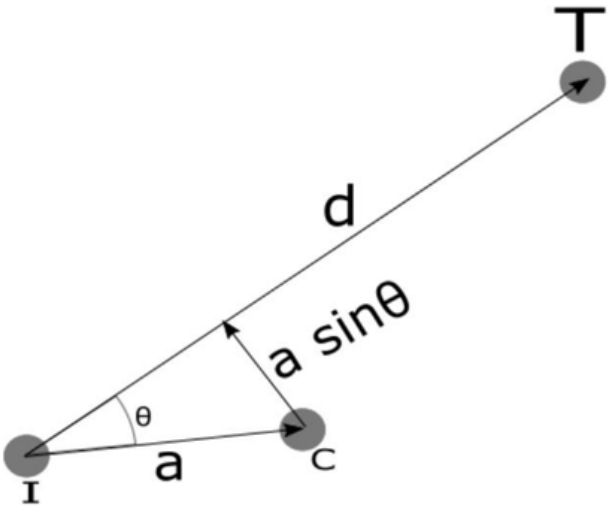

Figure 8: PD course correction:

(I – Initial Position, T – Target Position, C – Current Position)

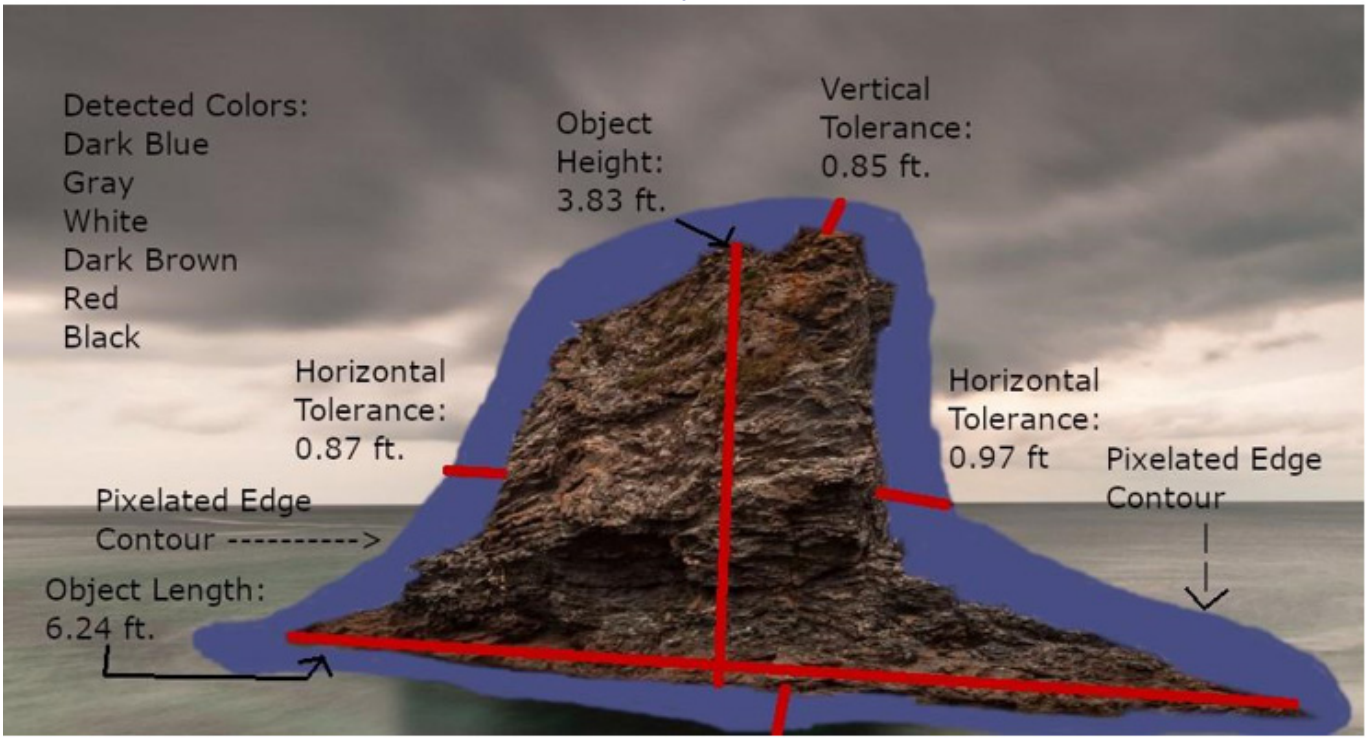

Figure 7: Detected Parameters in Computer Vision Software

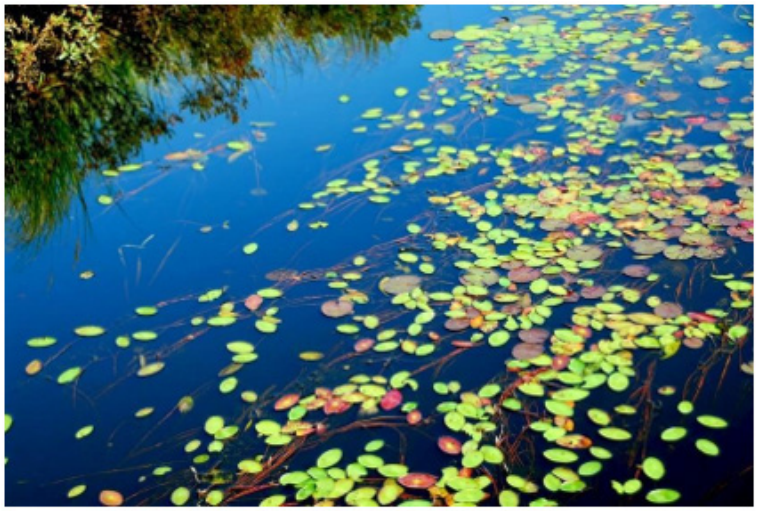

Figure 10: CV Test: Small weeds/grass

Figure 9: CV Test: Rocks in a lake

Figure 11: CV Test: External Boat

Figure 12: CV Test: Lifeguard/External Building

Figure 13: CV Test: Harmful/Domestic Animals

Figure 14: CV Test: Tree Branches

Figure 15: CV Test: Lilypads

Figure 16: CV Test: Swans/Birds

Figure 17: CV Test: Other Robots

Figure 18: CV Test: Plastic/Excess Debris

Table 1: GPS Coordinates of each waypoint for our autonomous algorithm applied to Lake Erie

	1	2	3	4	5	6	7	8	9	10	11	12	13
Lat.	41.7	42.0	41.4	42.1	41.6	42.1	42.6	42.2	42.0	42.5	42.2	42.8	42.4
Lon.	-83.3	-82.7	-82.5	-82.2	-81.8	-81.7	-81.4	-80.9	-80.5	-80.9	-80.2	-80.1	-79.7

Table 2: Length of Path Traversed and Rotational Angle (From Vertical) for each Waypoint Pair Line Segments

	1	2	3	4	5	6	7	8	9	10	11	12	13
Length (km.)	59.9	66.7	81.7	64.7	56.2	60.8	50.8	39.8	64.6	66.5	67.2	55.2	47.3
Angle (Deg.)	26.6	18.3	23.2	38.7	11.3	31.0	51.3	63.4	38.7	66.8	9.46	45.0	26.6

Table 3: Experiment 1, Computer Vision: a) Accuracy of Detection

Image		**Trial 1**	**Trial 2**	**Trial 3**	**Trial 4**	**Trial 5**
Large mass of rocks	Detection Goal: Success	Success	Success	Success	Success	Success
	Size of Rectangle (pixels)	199 x 154	171 x 153	320 x 214	217 x 145	193 x 154
Small weeds/grass	Detection Goal: Failure	Failure	Success	Failure	Failure	Success
	Size of Rectangle (pixels)	N/A	25 x 17	N/A	N/A	33 x 44
External boat	Detection Goal: Success	Success	Success	Success	Success	Success
	Size of Rectangle (pixels)	553 x 103	485 x 132	526 x 47	249 x 53	549 x 85
Lifeguard/External building	Detection Goal: Success	Success	Success	Failure	Success	Success
	Size of Rectangle (pixels)	394 x 203	320 x 334	N/A	432 x 220	460 x 189
Harmful/Domestic animals	Detection Goal: Success	Success	Success	Success	Success	Failure
	Size of Rectangle (pixels)	130 x 80	0 x 132	48 x 73	0 x 345	N/A
Tree branches	Detection Goal: Failure	Failure	Failure	Success	Failure	Failure
	Size of Rectangle	N/A	N/A	174 x 114	N/A	N/A
Lily pads	Detection Goal: Failure	Failure	Failure	Failure	Success	Failure
	Size of Rectangle (pixels)	N/A	N/A	N/A	352 x 201	N/A
Swans/Birds	Detection Goal: Success	Success	Success	Success	Success	Success
	Size of Rectangle (pixels)	247 x 120	134 x 143	35 x 125	37 x 116	139 x 105
Other robots	Detection Goal: Success	Success	Success	Success	Success	Success
	Size of Rectangle	336 x 81	306 x 83	331 x 108	327 x 49	319 x 139
Plastic/Excess debris	Detection Goal: Success	Success	Failure	Success	Success	Success
	Size of Rectangle (pixels)	562 x 276	N/A	536 x 258	611 x 391	430 x 321

Table 4: Experiment 1, Computer Vision: b) Robot motion based on vision

Image	Time for Detection (s)	Robot Turns (Yes/No)	Decision Correct (Yes/No)
Large mass of rocks	5	Yes	Yes
Small weeds/grass	10	No	Yes
External boat	3	Yes	Yes
Lifeguard/External building	4	Yes	Yes
Harmful/Domestic animals	9	No	No
Tree branches	8	Yes	Yes
Lily pads	12	No	Yes
Swans/Birds	7	Yes	Yes
Other robots	13	Yes	Yes
Plastic/Excess debris	7	Yes	Yes

Table 5: Experiment 2: PD Course Correction Control

Trial	Initial	Latitude	Longitude	PD	Azimuth Change	$a \times d$	$\cos\theta$	Displacement	Updated Location
1	Home	40.1290474	-83.1815978	Yes	Initial: -1.8029 Final: -0.3048	-6.68E-03	-0.887 Desired: -0.9	Latitude: 5.42E-05 Longitude: 1.041e-4	Latitude: 40.19010162 Longitude: -83.8184939
2	Patio	40.1290558	-83.1815246	Yes	Initial: 1.0624 Final: 0.4628	-1.13E-02	-0.1156 Desired: -0.13	Latitude: 9.1800e-5 Longitude: 1.07E-05	Latitude: 40.1291476 Longitude: -83.181514
3	Road	40.1292388	-83.1815075	Yes	Initial: 1.0297 Final: 1.0297	-2.26E-02	-0.0931 Desired: 0	Latitude: 1.83E-04 Longitude: 1.71E-05	Latitude: 40.1314006 Longitude: -83.183521