Real-World Testing of a Multi-Robot Team

Paul Scerri, Prasanna Velagapudi, Balajee Kannan, Abhinav Valada,
Christopher Tomaszewski, Adrian Scerri, Kumar Shaurya Shankar, Luis Bill,
and George Kantor

The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA
pscerri@cs.cmu.edu

Abstract. Multi-robot systems (MRS) have great promise for revolutionizing the way a variety of important and complex tasks are performed. While the underlying science is advancing quickly, the engineering problems associated with deploying multi-robot team under real world constraints have not been adequately addressed. In this work, we are developing teams of Cooperative Robotic Watercraft (CRW) for critical applications including flood response, water monitoring and security. This paper details the steps in the development and deployment of our low cost, robust CRW system, including design considerations, system description, user interface and subsequent field testing results. We took the watercraft into real environments and ran them through the types of exercises they will perform in real deployments, to better understand the full range of issues involved in creating and deploying real multi-robot systems. We report field testing results from three unique and different environments: four days of testing in an irrigation pond, six weeks in the Philippines, including after a typhoon and several hours of testing in a fish farm; resulting in more than 100 boat hours in the water and hundreds of thousands of data points. By the end, the process and the resultant boats were effective and robust, and could be controlled by one non-computer science undergraduate student and local Filipinos with no formal education.

Keywords: Multi-robot systems, Autonomous surface vehicle, Human robot/agent interaction, Flood disaster mitigation, Autonomous sampling

1 Introduction

Multi-robot systems (MRS) have received a great deal of attention recently due to their potential to address complex distributed tasks such as environmental monitoring, search and rescue, agriculture, and security[7–9, 2, 4]. Much research has been performed to develop the robots, algorithms, interfaces and concept of operations for a variety of multi-robot systems. One specific type of multi-robot system that has significant near term promise is fleets of autonomous watercraft. Small watercraft are an attractive option for real world multi-robot systems because some of the most critical robotic problems are minimized on water as movement is relatively simple and dangers are relatively low. However there have been relatively few efforts at using multi-robot teams in real environments, consequently the associated engineering issues for real-world deployment are ill-defined.

In this work, we address the engineering issues behind developing teams of Cooperative Robotic Watercraft (CRW) for applications including flood response, water monitoring and surveillance. We envision very large teams of CRW, perhaps numbering in the thousands, moving autonomously in large bodies of water under the supervision of a small number of operators attempting to achieve a complex task. Previous work has detailed the challenges involved in such coordination from a multi-agent perspective, including challenges in task allocation, information sharing and adjustable autonomy[12]. Putting fleets of boats out in water, in remote locations, will help clarify assumptions, change priorities and expose new issues for the community and help close the gap between the identified challenges and real-world deployment of such systems. The contribution of this paper is to describe in detail the process of developing a team of CRW for use in realistic environments, describe the field tests and disseminate the lessons learned.

The overarching goal of our work is to develop a low-cost multi-robot system that is easy to deploy and has sufficient robustness to make it feasible to deploy large teams in realistic environments with a reasonable amount of effort. This results in design constraints that are unusual to research robots. Section 3 outlines these considerations and describes the specific design choices we have made. The success of our approach has been validated through field trials, including a four day test at an irrigation pond in Maryland, a six week expedition to various locations in the Philippines and several hours of testing in a fish farm. A summary of these trials and associated experimental results are described in Section 4. Finally, Section 5 provides some commentary on the lessons we have learned in this process together with a description of our plans for future research.

2 Related Work

Most existing multi-robot systems use generic domain independent research platforms. While these are ideal for design, development and testing of associated software algorithms, they do not capture real-world constraints and are therefore not practical for deployment. Specialized robotic watercraft have been successfully used in deep sea tasks ranging from mapping deepest underwater caves [3] to tele-supervised sensor fleet for ocean surface and sub-surface studies [14]. Tele-supervised Adaptive Ocean Sensor Fleet [1] is an example of one such deep sea multi-robot science exploration system that combines a group of robotic boats to enable in situ study of phenomena in the ocean-atmosphere interface, as well as on the ocean surface and subsurface. The OASIS platform is a long duration solar powered autonomous surface vehicle, designed for autonomous global open ocean operations [11]. While these platforms are extremely capable and engineered specific to the requirements of the operating domain, the large associated cost with these platforms make them infeasible for large scale deployment.

Over the years, numerous architectures have been designed for multi-robot teams and challenges [5, 6, 10, 13, 15, 16]. There has also been some exciting work, developing MRS a wide variety of exciting capabilities [7, 8], but under tightly controlled conditions. However, typically it is only individual robots that have been evaluated under real-world conditions [9, 2, 4], hence the additional challenges for MRS are not yet fully understood.

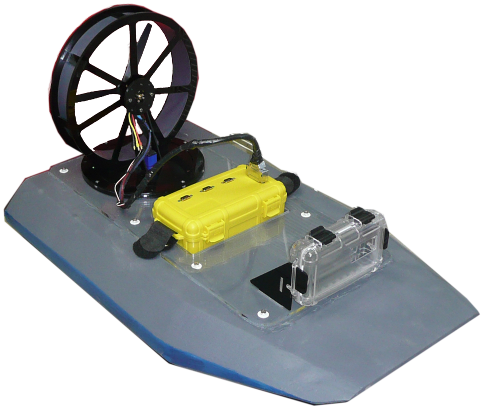


Fig. 1. A complete airboat.

3 System Description

We specifically chose to work with robotic boats, because they appear to be an ideal platform for looking at multi-agent issues within a real multi-robot system. Small autonomous boats do not pose the same danger to people and property that ground or air robots do. Compared to flying or traversing unknown terrain, moving in water is simple and safe. Although challenges arise dealing with currents and other water movement, these can be dealt with intelligently by planning and replanning, unlike many terrain features which can incapacitate a ground robot. Moreover, areas of water tend to be large, making them naturally suited to large robot teams that are of particular interest in the multi-agent community. Finally, water presents many interesting and important scientific, monitoring, disaster response and security problems that appear to be ideally suited to autonomous systems. For example, monitoring water quality over large areas is of scientific and policy importance but is prohibitively expensive with fixed sensors or manned survey vehicles. In the remainder of this section, we describe the design considerations and design for the major parts of the system.

3.1 Mobility and Propulsion

Design Considerations Given the constraints of field testing, we identified a set of essential criteria for platform development. These include the need to design and develop low-cost, robust, easy to manufacture and repair watercraft that are compact in size and weight and have a limited payload capacity.

Since the aim was to make large scale, low-cost robot teams, there were some unusual additional constraints on the physical designs of the robots. Not only did the robots need to be cheap, but the construction process had to be simple and fast, since the real cost of the robot would be high if many hours of construction were needed. Moreover, the robots needed to be designed to be highly robust, since for a large team even a relatively low failure rate would result in a cost prohibitive amount of time for repairs. However, some repairs are inevitable, hence the robots need to be designed to be taken apart easily and have components easily replaced.

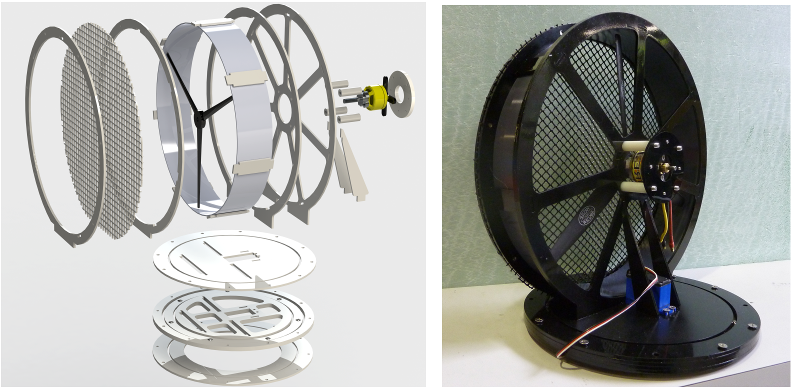


Fig. 2. Exploded view of the propulsion assembly (left) and a complete assembly (right).

Design We chose an airboat design (Figure 1), where the propulsion comes from a fan placed above water, for our watercraft platform for two important reasons. First, keeping the propellor above water is advantageous where the water might be shallow, e.g., in flooded environments or in ecologically interesting areas like reefs or estuaries. Second, the above water fan can be simply encased in a wire mesh for safety, making the boats safe for autonomous operation even around curious children.

Figure 3 shows the basic components of the airboat. A boat is approximately 70cm long and weighs about 4.4kg without batteries. Over the course of development and testing, we experimented with various configurations of batteries, with associated weight, cost and deployment time tradeoffs. In a usual testing configuration, a NiMh battery is used that weighs approximately 1.5kg and allows the boat to drive continuously at approximately 10km/h for a period of two hours. The choice of the size and weight for the boat were made to suit urban flood conditions, where safety and maneuverability are key requirements.

The hull is cut from sheets of insulation foam that were previously glued together and sealed with paints and sealants available at any hardware store. The fan and shroud assembly (Figure 2) is laser cut from extruded acrylic with PVC, stainless steel and polyethylene components. This assembly is an example of the design effort that needs to go into building cheap, robust components. We started with an initial design that was a scaled down version of a traditional airboat design, with a fixed fan and two rudders directing the air to steer the boat. However, the rudders were difficult to construct and often needed repairs. The current design uses a servo motor to actuate the fan to control the direction, and is more robust, efficient and improves the overall maneuverability of the boat. All aspects of the boat design went through many iterations, some iterations in design tools, often in physical designs, before a good design was reached – achieving simplicity was complex!

3.2 Computing and Electronics

Design Considerations Personnel who are not experts in robotics must be able to operate the robots since it will be impractical to require highly trained robotics experts for many potential applications. The procedures for transporting, starting, charging and maintaining the robots must be made simple and robust enough for non-experts. The technical infrastructure required for the robots must be minimized to maximize the range of environments where the robots can be used. A key problem is communication in large outdoor environments. Even during extreme events like floods and in underdeveloped parts of the world, cellular phone coverage is reasonably reliable and, thus, a feasible option. However, it will not always be possible to rely on phone networks, thus in certain situation the watercraft need to be able to provide their own communications facilities, e.g., by creating an ad hoc network, or operate without communication for short periods of time.

Design Rather than individually assembling a computing platform, a core design decision was to use a commercial smartphone like a Google Nexus to provide the computing, camera and communications for the boat. It is impractical to put together a similarly powerful, robust and tightly packaged custom computing platform at anywhere near the cost of a smartphone. Moreover, using a smartphone gives us access to multiple modes of communication, since most phones have WiFi, 3G and Bluetooth communication options. We chose Android based phones because of their relatively open and powerful development environment.

For communicating with sensors, motors and servos, we used an Arduino Mega, a relatively low-cost microcontroller board that provides a fast, flexible array of digital and analog I/O for controlling the fan shroud, gyros, and external sensors modules. The Arduino and smart phone communicate via Bluetooth,

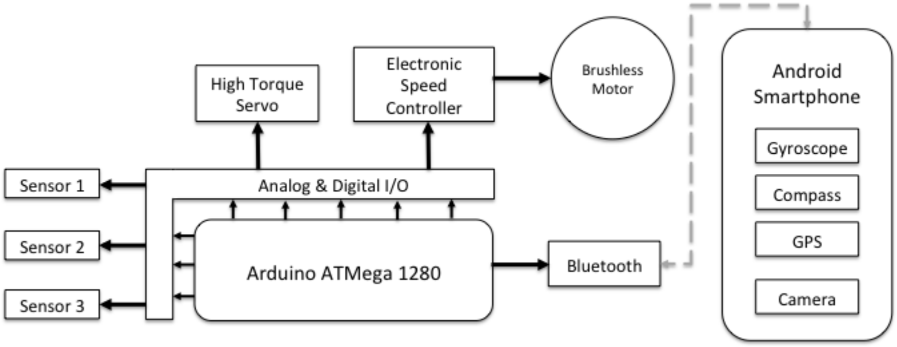


Fig. 3. Hardware functional diagram

which works extremely well over the short distance between the phone and Arduino. The servo for turning the fan, the fan itself and sensors are all connected directly to the Arduino which has a simple, high level protocol to the phone. External sensors are plugged directly into the Arduino using either digital or analog channels, depending on the sensor. The entire electronics assembly is encased in two waterproof boxes. One box contains the phone and is positioned near the front of the boat to best utilize the camera's field of view. The other box which is heavier since it contains the battery, is placed closer to the boat's center of mass.

A key design feature is the self containment of the major system components. A single baseplate holds the shroud to the hull, the two electronics boxes sit in precut locations and the fan assembly is simply screwed on. Each of these components can be simply lifted or screwed off and replaced within a couple of minutes.

3.3 Software

Design Considerations The primary issue to overcome pertains to the quality of obtained data from the embedded sensors required for control, specifically the GPS, gyro and compass. Layers of filters are required to smooth the data to extract sufficiently clean information to effectively control the boat. The software also had to be flexible to emerging needs, since the final application of the fleet is still not known. An early decision was made to have all the software be completely open source. This type of development requires some extra care in the architecture design, since there is less control over the implementation and use of specific components.

Design The overall system software is shown in Figure 4. The implemented software builds on the Robot Operating System (ROS), which provides a flexible publish-subscribe architecture with extensive built in debugging capabilities and a manageable development path. As noted above, the computation for the boat is provided by an Android smartphone and the local intelligence for each boat resides on the phone. Layers of functionality separate general modules from application specific modules. An end user interface provides a single operator with an overview of the state of the boats and provides high and low level commands for interacting with them. Each of the key components is described in more detail below.

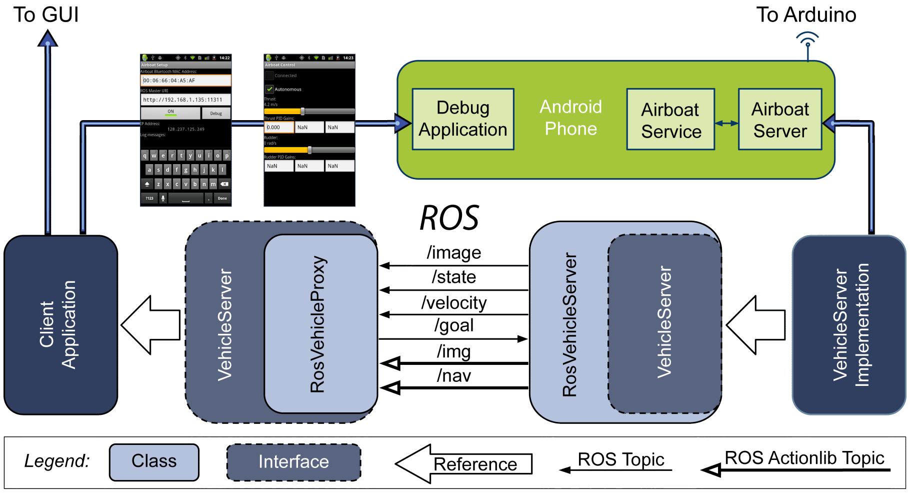
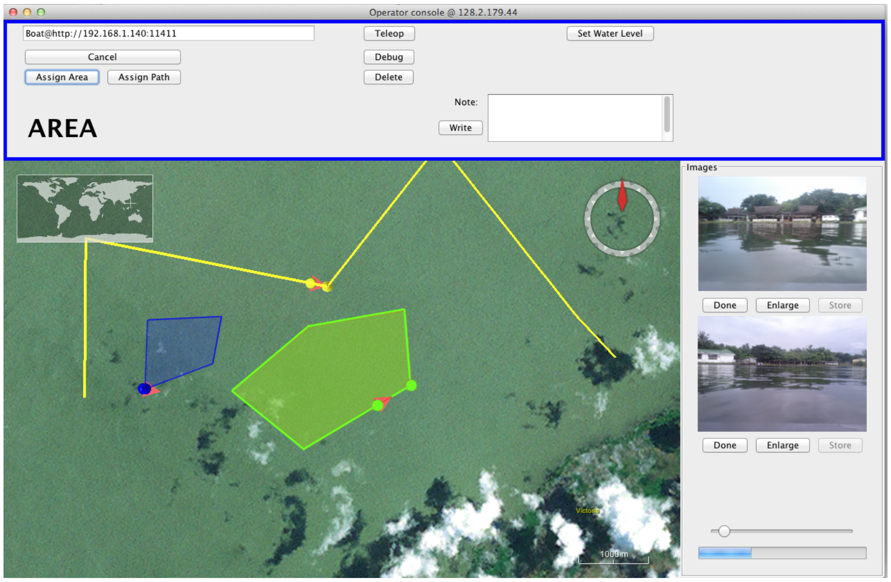


Fig. 4. The overall software architecture, showing the ROS components running on the Android phone, the connection to the Arduino Mega and the debug interface.

ROS is an open-source toolkit that provides libraries and tools to help software developers create robot applications. ROS has well defined hardware abstraction, support for device drivers, message parsing libraries and visualization tools. We use ROSJava, a pure Java based implementation of ROS that can be run directly on the phone. A ROS core keeps track of all Publishers and Subscribers that communicate information relevant to the reasoning of the system. Publishers and Receivers are modules (or nodes in ROS terminology) responsible for specific aspects of overall behavior. In initial tests, the ROS core was run on a remote machine off the boat, with only select modules actually running on the phone. Although communication intensive, this allows for ease of development and system evaluation, substantially reducing the overall development cycle. Later in development, the ROS core was moved to the phone and all processes were subsequently executed locally.

The boat executes the above described core functionality via a boat server. Client applications and additional modules running on the phone provide do-


Operator console @ 128.2.179.44

Boat@http://192.168.1.140:11411

Fig. 5. Screenshot of the operator console with three boats in Laguna Lake.

main specific functionality that leverage the core functionality by other ROS modules. This design allows us to make subtle changes for specific domains without impacting previously tested and reliable code. For example, the boat behavior required when it loses communication with the base station will vary significantly depending on the domain. During testing, the boat should attempt to go back to home base whereas for a water sampling domain it might return to communication range only periodically and for a flood response domain it should return regularly to provide data to first responders. This domain specific logic is captured in the client applications without adversely affecting the core functionality that implements the actions.

The top level intelligence of the boat, the reasoning about where and what the boat should do is encapsulated in a proxy. Currently the proxy runs on an operator's machine and has relatively low overhead in terms of communication with the boat. However, once the reasoning is reliable, the proxy will reside on the phone and interact closely with the ROS software. Currently, the implemented proxy is responsible for path planning to implement high level operator directives about areas to visit or search. Additionally, we have designed and tested an initial adaptive sampling proxy to allow the boats to sample areas of highest uncertainty when building maps of defined water property.

A centralized user operator interface provides the operator with enhanced situational awareness about the multi-robot teams and the operating domain (Figure 5). The interface provides information about the locations of all the boats, overlaid on a map of the environment. Using the interface the operator can specify high-level objectives either as waypoints, paths or areas to search, or low-level direct commands to the boats. The watercraft will send back images from the on-board camera at approximately 1 Hz. An image queue on the operator's side receives and reorders these images for the user, allowing them to observe, discard or save images for later use. The operator interface emphasizes simplicity and reliability over complex functionality.

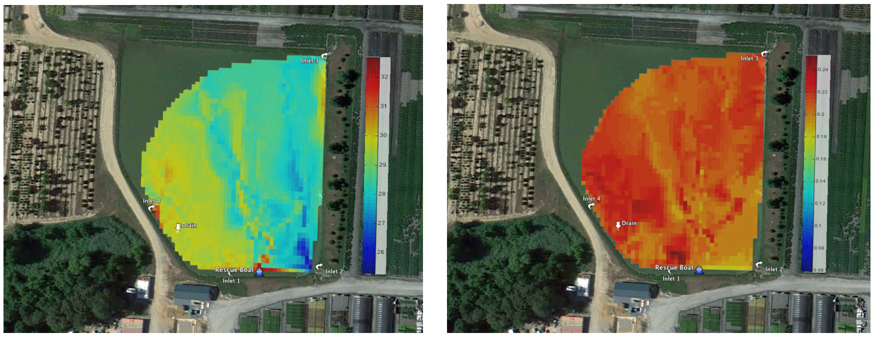
Lessons Learned Despite the promise of ease of use and ubiquity, unfortunately, not every aspect of working with an Android smartphone as a computational platform has been positive. We have implemented vision based obstacle avoidance that segments out water to identify obstacles. However, the current Android operating system provides limited low level control over phone functions and thread execution. This consequently increases the computational overhead and to date has proved impossible to get images and run obstacle avoidance processing without unpredictable and unacceptable interactions with other reasoning. Perhaps more critically, service providers limit socket based access to 3G phones in very provider specific ways, making using 3G infeasible. Hence, we are currently limited to using wireless communication.

4 Field Trials

To evaluate the developed CRW system under real-world conditions, we deployed them at three separate sites and put them through a series of tests. First, we took a team of three watercraft to an irrigation pond at a large nursery to sample the water conditions across the pond. The test identified issues with the deployment and use of the boats that were addressed prior to second field deployment. In the second deployment, a team of five boats were sent to the Philippines to learn more about deploying in areas without the infrastructure we typically rely on. Five boats were deployed multiple times in small areas, sometimes under the control of locals without formal computer science training. The boats were predominantly used for water sampling but were also briefly evaluated in the aftermath of a typhoon. In the third deployment, a team of four boats were used to analyze the levels of dissolved oxygen in a fish farm. The test identified the observed oxygen values change very slowly, but very consistently to changing conditions.

Four days testing in the irrigation pond, six weeks in the Philippines (one week intensive and ongoing occasional testing) and several hours of testing in the fish farm has resulted in the boats being taken out about 20 times accounting to more than 100 boat hours in the water, tens of kilometers covered and hundreds of thousands of data points. While initial testing was slow, frustrating and involved a lot more time with the boats out of the water than in, by the end the process and boats were sufficiently usable and robust that one non-computer science undergraduate student and local Filipinos with no formal education were able to deploy and use the boats. In fact, one of the biggest surprises was the comfort of local Filipino people with the technology and the speed at which they were able to familiarize themselves with it. By far the biggest problem encountered was with wireless communication, with the real-world details of various wireless technologies, particularly 3G causing difficulties.

4.1 Moon Nursery Pond Test



(a) Temperature

(b) Electrical conductivity

Fig. 6. Results from the Moon Nursery irrigation pond. Variation observed across the pond was the result of drains bringing water from different fields.

The first field testing was done at an irrigation pond at a nursery. This pond is scientifically interesting because the nursery recycles the water, spraying the plants with water from the pond then capturing the run-off back in the pond. This approach is environmentally exciting, as it reduces water waste, but there is a concern regarding water quality over time due to accumulating fertilizers and pesticides. Biologists have two stationary buoys in the pond, measuring various properties of the water. We deployed three boats out at the lake over four days of testing. A key aim was to sense across the whole pond, to interpolate between the data collected by the biologists. We used sensors that measured electrical conductivity, a property of water that correlates well with the *total dissolved solids* in the water, a key measure of interest to scientists as well as temperature and pH.

Figure 6 shows a plot of the electrical conductivity and the temperature across the pond, as measured by the boats. Notice that both measures vary significantly across the pond, with the scientist's fixed buoys (which were placed near the top right and bottom left of the pond) giving only part of the picture. This shows the value of using mobile sensors like watercraft to sample the pond. During this test, we tried simple sampling patterns (primarily a lawnmower pattern) and a simple adaptive sampling algorithm. The adaptive sampling algorithm would send the boat to the location where previous readings had shown maximum uncertainty, intuitively attempting to minimize overall uncertainty as quickly as possible. However, it turned out that uncertainty was relatively uniform across the pond and the adaptive sampling worked qualitatively the same as the simple patterns.

4.2 Philippines Test

In September 2011, two undergraduate students and a recently graduated Masters student took five boats to the Philippines. They were joined by observers from the University of the Philippines and from local aid organizations. Primary testing lasted for one week, after which two of the students returned home leaving one (non-CS) undergraduate student to continue testing. Testing was performed in several locations including Laguna de Bay, Taal volcano, a village during flooding in the aftermath of twin typhoons and a fish farm. A key aim was to have all five boats in the water at the same time, under the control of the same operator. This was achieved a number of times. In total there were more than 15 tests in seven different locations.

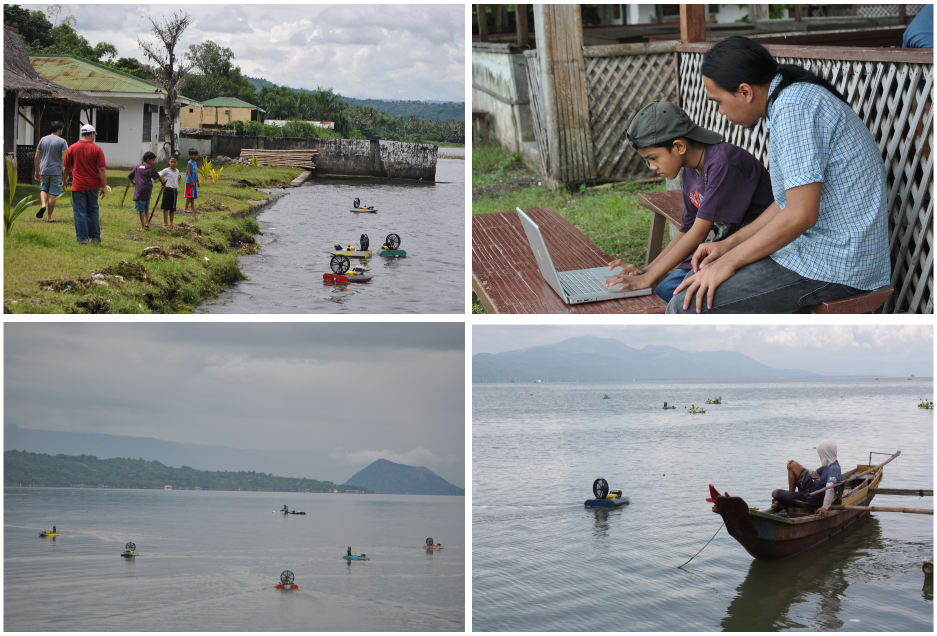


Fig. 7. Our six week deployment in the Philippines demonstrated an ability to deploy five airboats simultaneously in remote locations with a control interface simple enough to be used by a child.

Some key tests are summarized below.

- *September 7th* Initial test under manual control in Laguna Lake. Winds cause significantly larger waves than the boats had encountered before, which reduced performance but the boats were still able reach their assigned destinations. The boats were controlled manually. Power for the base station was provided by running a power cord from a nearby house.

- *September 9th* First autonomous tests with three boats running simultaneously in Laguna Lake. Low winds allowed much better boat performance. Boats were directed to either go to waypoints, follow paths or traverse areas that the operator specified on the interface.
- *September 11th* Five autonomous boats simultaneously under the control of a single operator in Lake Taal. Used sensors to create electrical conductivity and temperature maps of the water around fish farming. A nine year-old Filipino boy, competently controlled three boats via the interface.
- *September 30th* A single boat was manually driven around flood water in Malabon resulting from Typhoon Pedring. The water was approximately 10cm deep. Many images were taken from the onboard cameras for testing future obstacle avoidance algorithms.
- *October 4th* A single boat autonomously drove around a fish farming pond in Dagupan. The sensors found this water to be the lowest in electrical conductivity, a proxy measure for *total dissolved solids*, of all the test sites.

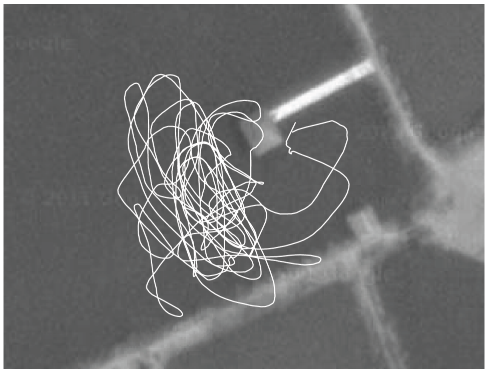


Fig. 8. Airboat trajectory of a single airboat operating in a fish farming pong in Dagupan.

Figure 8 shows the path taken by a boat at the fish farm, an interesting environment because of the complexity of the water and the need to keep the water healthy. Figure 9 shows a plot of the water temperature in the lake inside Taal volcano immediately before (left) and after (right) rain. This lake is important because a recent unexpected, rapid and significant rise in temperature caused $1.3M in losses to fish farming in the lake. The plot shows considerable variation in the temperature and significant differences due to the rain.

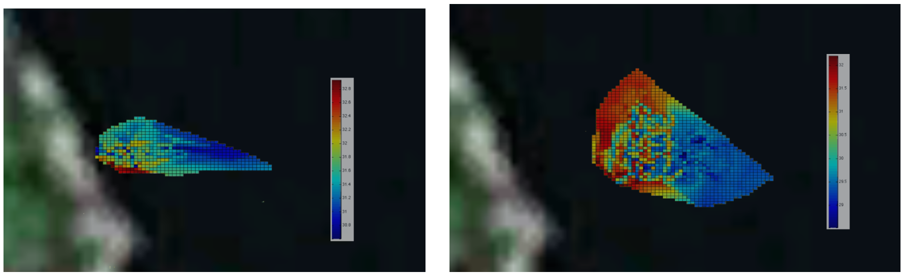


Fig. 9. Plots of temperature in Taal Lake before (left) and after (right) a tropical rain storm.

4.3 Fish Farm Test

The final field testing was done at a commercial fish farm. These farms spend considerable time and effort checking the levels of dissolved oxygen in their fish ponds. Whenever the level falls below a fixed value, aerators are turned on to add oxygen to the water. If dissolved oxygen levels fall for too long without being corrected, fish growth is stunted and fish may even die.

Our testing with water sensors at the Shelby Fish Farm showed that sensors measuring the water did not have the types of error that we originally expected. Specifically, there was very little random noise in the measurements. Instead, the error was dominated by hysteresis in the sensors as they adjusted to local water properties. Figures 10 show readings taken in each 10m x 10m area at the pond. The readings are organized in the order they were taken, but there may be temporal gaps in the sequence as the boat went to another 10m x 10m area. When readings were taken one after another within the same area, they were taken 1s apart. Notice that the values change very slowly, but very consistently. The figures show readings from a dissolved oxygen sensor, but values from temperature and conductivity sensors are qualitatively similar, but do react more quickly to changing conditions. This data has caused us to reevaluate our approach to searching the water. It is impractical to simply wait in each area until the data stabilizes, since this can take minutes. Instead, we are planning to use the derivative of the sensor data to bound the possible range of values in a particular area.

5 Conclusions

This paper describes the development of a team of low-cost Cooperative Robotic Watercraft (CRW), the associated engineering issues and early results in deploying a small numbers of these boats under real-world conditions. The boats and software were designed to be cheap, robust and easy to build and maintain. Successful trials in a fish farm, an irrigation pond and in the Philippines showcase the utility and usefulness of the developed MRS. The aim of this testing was

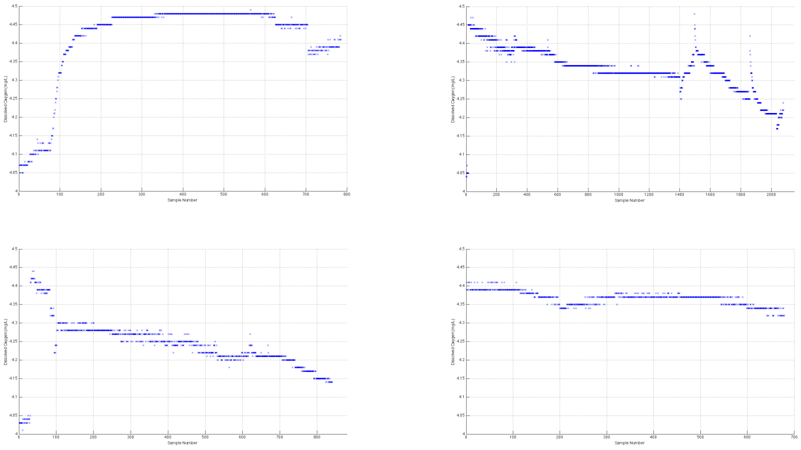


Fig. 10. Plots of dissolved oxygen content collected at Shelby Fish farm.

not to make any specific technological breakthrough or evaluate any particular algorithm, but to better understand the challenges of deploying real MRS. The following lists summarizes some of the key lessons learned in the testing.

- *Design for Openness* We encountered various issues ranging from logistics to design that had to be overcome for successful deployment, e.g., one part of the fan assemblies was lost in transport. Fortunately, the design was open and simple enough that a trip to a local hardware store and some improvised cutting was enough to replace the parts. We were fortunate that the lost part could be replaced, not all components of the boats could have been. Future design iterations will aim to have even more components that could be quickly replaced if broken or lost.
- *Communication is a Problem* During field trials communication with 3G turned out to be infeasible. Ad hoc networking appears to be the logical approach, but reliable, usable packages for Android are not readily available. Perhaps more importantly, the intelligent reasoning to create and use an ad-hoc network while executing a primary mission and having operators control some of the robots does not exist. This needs to be a priority research issue for real world MRS.
- *Comfort with Technology* We were pleasantly surprised by the comfort of untrained local people with the robotic technology and how quickly they were able to operate the robots. As an extreme example, Figure 7 (top, right) shows a nine year old boy sending the boats around part of Laguna de Bay. While we typically think of robots as something requiring expert training, if things are kept simple, graphical and intuitive, people that have grown up with technology can quickly learn. This is exciting for real applications and perhaps has implications for interface design.

- *Unknown Killer Apps* The boats were initially designed for flood response and environmental water monitoring. However, actually taking the technology out and showing it to people working in the environment led to suggestions for new applications that may actually be more realistic in the near term than ones we had envisioned. Local government officials and environment policy officials suggested applications including surveillance for illegal logging, fishing and polluting and monitoring water in fish farms. The lesson here is that getting the basic technology working and into the hands of the people that understand the real problems can be the best way of working out how to use the technology.

Many important technical lessons were learned, both positive and negative justifying the effort that went into performing the field tests. The deeper lesson is that MRS are rapidly maturing to the point where we can seriously think about using them for real world applications. Ongoing work is focused on two specific issues, highlighted by the testing. We are looking at reasoning about communication, dealing with and creating ad-hoc networks and being intelligent in the face of communication disruptions. Secondly, we are looking to develop adaptive sampling techniques that find and focus on features that might be of interest to scientists.

References

1. A. Elfes, G.W. Podnar, J.M. Dolan, S. Stancliff, E. Lin, J.C. Hosler, T.J. Ames, J. Higinbotham, J.R. Moisan, T.A. Moisan, et al. The telesupervised adaptive ocean sensor fleet (taosf) architecture: Coordination of multiple oceanic robot boats. In *Proc. IEEE Aerospace Conference*. Citeseer, 2008.
2. H. Endres, W. Feiten, and G. Lawitzky. Field test of a navigation system: Autonomous cleaning in supermarkets. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1779–1781. IEEE, 1998.
3. N. Fairfield, G. Kantor, D. Jonak, and D. Wettergreen. Autonomous Exploration and Mapping of Flooded Sinkholes. *The International Journal of Robotics Research*, 29(6):748, 2010.
4. S. Hayashi, K. Shigematsu, S. Yamamoto, K. Kobayashi, Y. Kohno, J. Kamata, and M. Kurita. Evaluation of a strawberry-harvesting robot in a field test. *Biosystems Engineering*, 105(2):160–171, 2010.
5. N. Jennings, E. Mamdani, I Laresgoiti, J. Perez, and J. Corera. GRATE: A general framework for cooperative problem solving. *Intelligent Systems Engineering*, 1(2), 1992.
6. Zhiang Lin and Kathleen Carley. Dycorp: A computational framework for examining organizational performance under dynamic conditions. *Journal of Mathematical Sociology*, 20, 1995.
7. K.H. Low, G. Podnar, S. Stancliff, J.M. Dolan, and A. Elfes. Robot boats as a mobile aquatic sensor network. In *Proc. Workshop on Sensor Networks for Earth and Space Science Applications (ESSA) at the International Conference on Information Processing in Sensor Networks*, 2009.

8. G.A. Oosthuizen, A. Al Shaalane, et al. Evaluation of existing robot technologies for deep level mining applications. In *ISEM 2011*, 2011.
9. Charles L. Ortiz, Regis Vincent, and Benoit Morisset. Task inference and distributed task management in centibots robotic systems. In *AAMAS*, 2005.
10. Richard Pew and Anne Mavor, editors. *Modeling Human and Organizational Behavior*. National Academy Press, Washington, D.C., 1998. National Research Council.
11. G.W. Podnar, J.M. Dolan, A. Elfes, S. Stancliff, E. Lin, JC Hosier, T.J. Ames, J. Moisan, T.A. Moisan, J. Higinbotham, et al. Operation of robotic science boats using the telesupervised adaptive ocean sensor fleet system. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1061–1068. IEEE, 2008.
12. P. Scerri, B. Kannan, P. Velagapudi, K. Macarthur, P. Stone, M.E. Taylor, J. Dolan, A. Farinelli, A. Chapman, B. Dias, et al. Flood disaster mitigation: A real-world challenge problem for multi-agent unmanned surface vehicles. In *AAMAS'11 Workshop on Autonomous Robots and Multi-robot Systems*, 2011.
13. Paul Scerri, David Pynadath, and Milind Tambe. Don't cancel my barcelona trip: adjusting the autonomy of agent proxies in human organizations. In *Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents — the human in the loop*, pages 169–173, 2000.
14. R. Smith, Y. Chao, B. Jones, D. Caron, P. Li, and G. Sukhatme. Trajectory design for autonomous underwater vehicles based on ocean model predictions for feature tracking. In *Field and Service Robotics*, pages 263–273. Springer, 2010.
15. M. Tambe, D. Pynadath, and N. Chauvat. Building dynamic agent organizations in cyberspace. *IEEE Internet Computing*, 4(2):65–73, March 2000.
16. C. Topper and K. Carley. A structural perspective on the emergence of network organizations. *Journal of Mathematical Sociology*, 24(1), 1999.