Proceedings of the 2006 IEEE International Conference on Robotics and Automation
Orlando, Florida - May 2006

Data Muling over Underwater Wireless Sensor Networks using an Autonomous Underwater Vehicle

Matthew Dunbabin and Peter Corke
Autonomous Systems Laboratory
CSIRO ICT Centre
PO Box 883, Kenmore QLD 4069, Australia
firstname.lastname@csiro.au

Iuliu Vasilescu and Daniela Rus
MIT Computer Science and Artificial Intelligence Laboratory
Cambridge MA 02493, USA
{iuliuv,rus}@csail.mit.edu

***Abstract*— We present algorithms, systems, and experimental results for underwater data muling. In data muling a mobile agent interacts with static agents to upload, download, or transport data to a different physical location. We consider a system comprising an Autonomous Underwater Vehicle (AUV) and many static Underwater Sensor Nodes (USN) networked together optically and acoustically. The AUV can locate the static nodes using vision and hover above the static nodes for data upload. We describe the hardware and software architecture of this underwater system, as well as experimental data.**

***Index Terms*— Underwater sensor networks, AUV, vision-based navigation, underwater robotics.**

I. Introduction

We wish to develop small autonomous underwater robots that are cooperative, adaptive, and can establish ad-hoc underwater networks. Such robots will permit the exploration and monitoring of underwater environments, allowing applications such as long-term monitoring of underwater habitats, monitoring and surveillance of ports, underwater geochemical prospecting and modelling the impact of weather, and ground activities (such as manufacturing and agriculture) on the water quality. Each of these applications requires long term underwater presence over a large area and agile response to triggers within the environment. The response might include visitation by a sensor-rich robot, data upload, device parameter (eg. sampling rate) change or reprogramming or even physical repositioning of the network nodes.

To perform such tasks, there needs to exist a synergy between mobility and communication. Sensor networks provide robots with faster and cheaper access to data beyond their perceptual horizon. Conversely, robots can assist a sensor network by deploying, moving and retrieving nodes, by localizing network elements post deployment, by making repairs or extensions as required, and acting as *data mules* to relay information between disconnected sensor clusters. The main aspects of these interactions are (1) locating and docking with modules; (2) placing, retrieving, and organizing modules; (3) cooperative navigation with docked AUVs; (4) communications, and (5) data muling over a deployed sensor node.

In this paper we describe our system, algorithms, and experiments for underwater data muling. The data mule is an Autonomous Underwater Vehicle (AUV) called *Starbug* and the underwater sensor nodes are called *Aquaflecks*. Together they form a network using two communications modalities: acoustic communications for broadcast and event signalling, and optical communications for high-rate local data transfer. Given a set of deployed underwater sensor nodes functioning as waypoints, the robot computes a tour that visits all the nodes. The execution of the tour consists of travelling to the next waypoint (using visual odometry), locating visually (using color object recognition) the sensor node, establishing optical communication with the node, and hovering while transferring the data from the sensor (using the optical communication system). We describe the algorithms for each of these tasks and present experimental results with a system consisting of Starbug and up to eight *Aquaflecks* conducted in a pool.

A. Related Work

There has been much work in the fields of AUVs, their control and navigation, underwater communications, sensors and docking [1]. In a 2000 survey [2], it was estimated that over 1,000 robotic underwater vehicles operate worldwide in industry, military and research applications. A small but growing portion of these are semi-autonomous or autonomous robots.

AUVs in general face severe navigation challenges since GPS is not available underwater. There have been three types of navigation systems for underwater robots, which rely on different sensors: (1) dead-reckoning and inertial navigation, (2) acoustic, and (3) geophysical navigation [3]. Visual feature tracking has also been employed, in particular on the Kambara project [4]. Others have found that combining sensor information, such as conventional long baseline acoustic sensing and Doppler velocity measurements, can improve underwater navigation [5]. Optical guidance systems, which give precise resolution at very short distances, have been used for the AUV's close-range homing and docking abilities. In [6] an optical quadrant tracker locks onto a light source for

0-7803-9505-0/06/$20.00 ©2006 IEEE

docking, as demonstrated underwater on a SeaGrant Odyssey IIB [7]. Alternatively, long baseline and ultra-short baseline acoustic beacons have also been used for docking [8].

Recently there has been interest in deploying multiple robots, in particular for oceanographic research applications. The Serafina project [9] explores large-scale formation control issues with multiple small, agile AUVs. Gliders such as Seaglider [10] are designed to dive to a pre-programmed depth and resurface whilst taking scientific measurements. These robots are capable of travelling thousands of kilometres in a triangular depth profile. Gliders have also been used in cooperative multi-AUV control research [11]. Such research is motivated by collaborative oceanographic research projects such as the Autonomous Ocean Sampling Network II [12]. It is becoming more important for the robots to be able to assist in the deployment of, or to act as parts of, such large-scale data-collecting networks. A small submarine [13] has been proposed as a sensor in such a network. The robot houses a Mote sensor and can control its own depth. In addition, some attempts have been made to create modular underwater robots. Inspired by eels [14] or lampreys [15], these are smaller-scale biomimetic robots whose modules are permanently joined in one configuration.

II. System Description

A. Underwater Sensor Node: the Aquafleck

We have constructed 20 underwater sensor nodes called *Aquaflecks*. Each node is built around a wireless sensor device developed by the CSIRO called a *Fleck* [16], based on an 8-bit CPU with limited memory. Each node has 512kbytes of flash memory for data storage. The Fleck has a custom daughter board for optical communications that drives a high-power LED and a sensitive matched photodiode. The LED emits light in a 30 degree cone and supports a maximum data rate of 320kbits/s at a maximum range of 2m (8m when augmented with a lens). A low-cost/low-power acoustic communication module using 30kHz FSK/PPM modulation with an omnidirectional range of 20m and a data rate of 50bit/s is also used for ranging. Each node has a pressure sensor, temperature sensor, and some have a CMUCam camera. All this hardware is encased in a water tight bright yellow Otter box. At the top of the box there is a 170mm rod with an LED beacon, which an AUV can use to locate the box, dock, and pick it up. The box also contains three C-cells that can power the node for several days when using continuous power, or several months on power saving modes with minimal beacon and communications usage.

B. Starbug AUV

Starbug is a small hybrid AUV designed for operating in highly unstructured environments [17]. An onboard PC/104 computer stack with a Crusoe processor running Linux performs vehicle control as well as all image processing from its stereo camera pairs for its powerful onboard vision-based motion estimation system [18].

Starbug's key specifications are: mass 26kg, length 1.2m (folding to 0.8m for transport), maximum forward thrust 20N, and maximum endurance of 3.5 hours (8 km at 0.7m/s) with current lead-acid battery technology. The vehicle can be fully actuated with six thrusters providing forward, lateral and vertical translations as well as yaw, roll and pitch rotations. Figure 2 shows the Starbug AUV as configured for these data muling experiments, minus the lateral tail thruster, and with the communication *Aquafleck* node located on the front of the vehicle pointing downward.

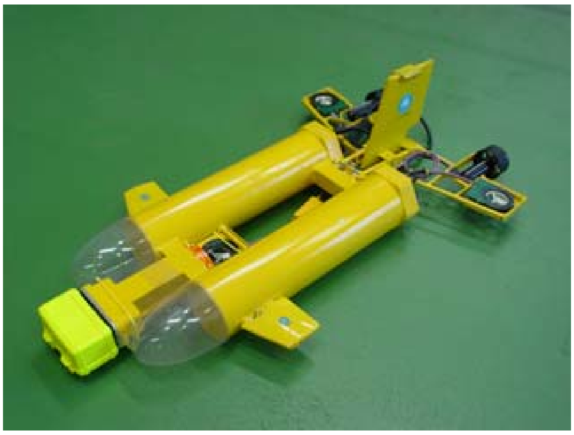


Fig. 2. The Starbug AUV.

Vehicle control software effectively decouples the thruster forces and allows independent control of vehicle forces and moments in 6DOF. All the thrusters are daisy-chained on a CANBus control network which allows for a single hull penetration. Internal sensors such as pressure and an IMU are also on the CANbus. Starbug has two stereo vision heads. One looking downward for estimating sea-floor altitude and speed as well as mapping, and the other looking forward for obstacle avoidance. The forward cameras were not used during this investigation.

The vehicle's position in Cartesian coordinates is estimated using the onboard visual motion estimation system described in [18]. This system has demonstrated position estimation to within 5% of distance travelled for transects of over 50m in length.

C. Communication and networking

Our approach to communication is based on a hybrid design which incorporates both optical and acoustic systems. The optical system is used for short-range (up to 8m) line-of-sight data transfer and communication between a sensor node and an AUV acting as a data mule (with data rate of 320 kbits/s). The AUV moves through the network uploading the stored data from the sensor nodes, and downloading commands. The acoustic system is used to signal events and

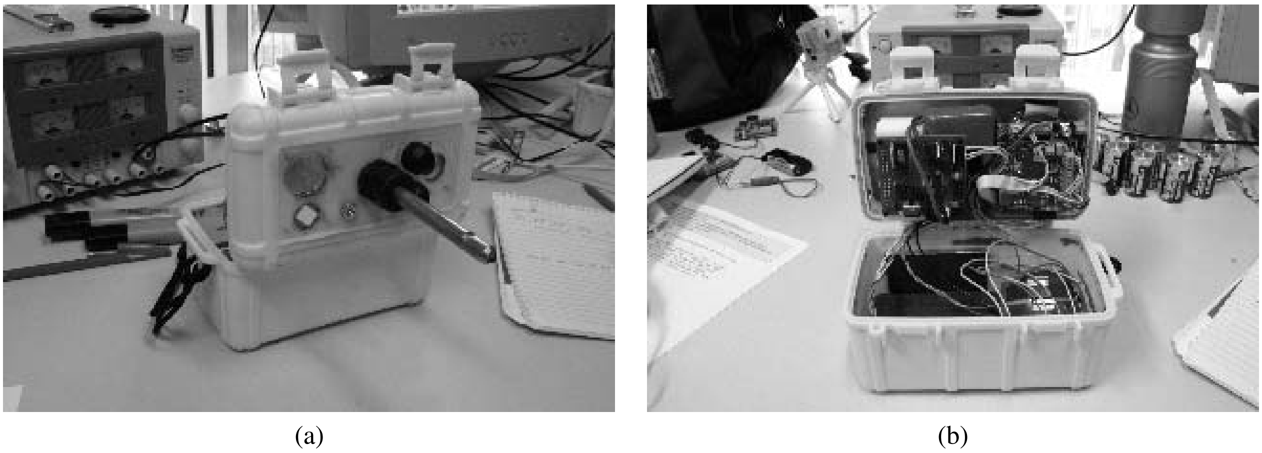


Fig. 1. The underwater sensor node. (a) Top of the node containing the sensors and the docking rod, (b) inside showing circuitry.

transmit small amounts of data. Signalling an event allows the AUV to move to the area of interest, and possibly redeploy the sensor network to concentrate on some important feature in the environment.

III. Algorithms

Data muling proceeds in several stages as follows:

1) Compute a route that goes through all the nodes.
2) Select the next node to be visited.
3) Travel to the approximate location of the selected node.
4) Locate the node visually and establish communications.
5) Hover while uploading the data.
6) If not the last node, go to (2).

The next sections detail our algorithms for the control algorithms that accomplish data muling.

A. Selecting the Next Node

Planning the sequence for data muling depends on how much prior knowledge of the sensor locations is available. It will be assumed that the sensor location map is made available to the robot, however, this map may not be of high accuracy. It is also assumed that all nodes have the same priority in terms of visitation.

Given a map with known locations any algorithm that approximates a solution to the travelling salesman problem can be used. However, since the underwater robot is subject to significant error and uncertainty while travelling underwater, we choose a method that allows the robot to minimize the distance to be travelled between nodes as much as possible. This may increase the total path length to visit all the nodes (and hence the endurance requirements of the AUV), however, with this method the robot will accumulate less position error and have an improved chance of finding a target node. The robot will reset its position estimate when it finds (and identifies) the node so we are concerned only with error accumulation along each segment of the tour.

The next node to be visited is the closest unvisited node. If the robot has a map of the node locations this next node is easy to determine. In the absence of a map, a communication protocol can establish the identify and location of this node.

B. Navigating to the Next Node

If the location of the next node is nearby, the robot navigates using its current location estimate and visual odometry [18] along with a magnetic compass to get there. If the node is far, the robot performs a GPS based transect whereby the vehicle periodically surfaces to obtain a GPS fix and correct its trajectory. This is repeated until the vehicle is within a specified radius of the desired node and then travels on the surface using GPS to locate itself over the node. A quick dive to the bottom reduces the navigation problem to the previous case. Figure 3 shows the case of a GPS based transect where the location of the first node is to the upper right corner of the figure.

C. Locating a node

A critical part of our experiments is the ability for a vehicle to identify and manoeuvre with respect to a node. Once a node is located, the AUV can perform data muling or docking and transport.

To simplify the task of locating nodes underwater we chose a passive method where the nodes are identified based on their color, bright yellow ABS plastic. This requires no energy expenditure on the part of the node as opposed to active beacons. It is also highly desirable as the Starbug AUV already has down-looking color cameras. A typical Starbug image of a node is shown in Fig. 4(a).

Our approach to locating the nodes by color is classical. The color images are converted to normalized red-green chromaticity coordinates and applied to a pre-learnt 2D lookup table, Fig. 4(c), which maps the pixels to a binary image, Fig. 4(b). Connected region analysis and an area threshold determine the presence of a node. Typical results

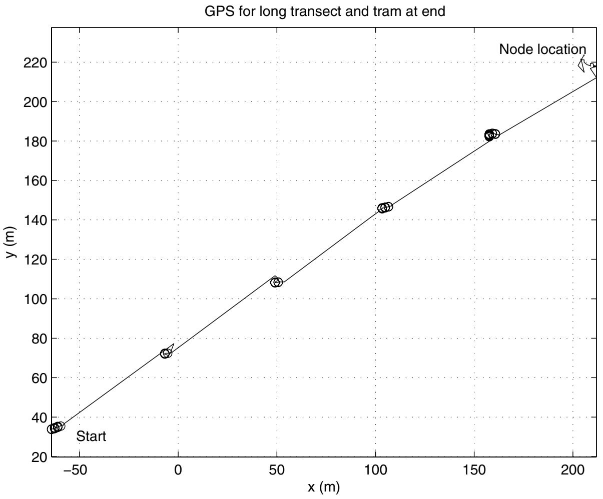


Fig. 3. GPS based transect to node location (circles indicate vehicle has surfaced to correct trajectory).

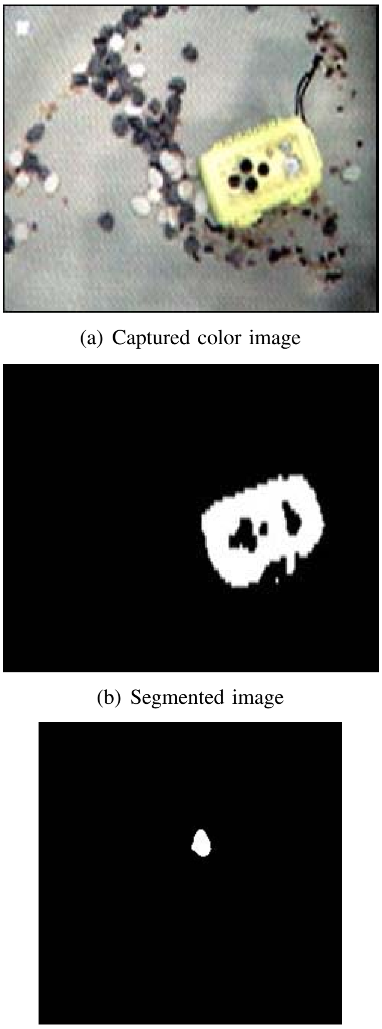
for centroid and blob area are shown in the lower 3 traces of Fig. 7. Performing this process with both the left and right camera, we can compute an approximate range based on the centroid disparity, and this allows for adjustment of the area threshold. In our experiments in several different environments nearly 200 nodes were detected successfully using this method .

Several factors combine to make this a challenging problem. Firstly, a significant area of the top face of the node is not yellow due to lenses, acoustic transducers and sensors. (see Fig. 4(a)). The preferential absorption of long wavelengths (reds) in water causes marked color change over quite short distances. This filtering occurs in the illumination falling on the node, and also along the path from node to camera. For longer term immersion, obscuration due to biofouling by marine flora and fauna may also occur.

If the robot does not initially locate a node when arriving at where it thinks one should be, a spiral search is initiated. In this search, the robot navigates along a spiral centred around the best approximate position of the *Aquafleck* using visual odometry to keep track of the position and prevent drifting due to currents. We implemented a constant pitch spiral whose centre is located at (x_0, y_0) such that the desired position at time t is:

$$(x(t), y(t)) = x_0 + t\ \cos(w\ t), y_0 + t\ \sin(w\ t)$$

where the constant w determines the pitch. The pitch is set by the field of view of the camera (consecutive passes should slightly overlap). This spiral is discretized and transformed into waypoints which are given to the robot's controller. Each time the robot is within a specified radius of the current waypoint, the next waypoint is set. If the node is not found after a specified amount of time, the robot surfaces, recalibrates its position, and tries again.


(a) Captured color image

(b) Segmented image

(c) R-G lookup table

Fig. 4. AUV image processing.

This spiral search method was found to be effective in locating "lost" nodes. Figure 5 shows the results of a vision-based spiral search in the presence of a significant water current from left to right. Here the robot's initial starting position meant that it missed the node by approximately 1.5m before the spiral was initiated. This spiral took 110s to complete.

D. Data Muling: Hovering

Once a node enters the vehicle's field of view, its current position is recorded and the robot switches to a station keeping mode while attempting to signal the node and upload data. The position of the node is recorded since, due to the relatively narrow field of view of the cameras, it is possible for the vehicle to lose the node whilst trying to stop or turn towards the node. Therefore, whilst the node is in view,

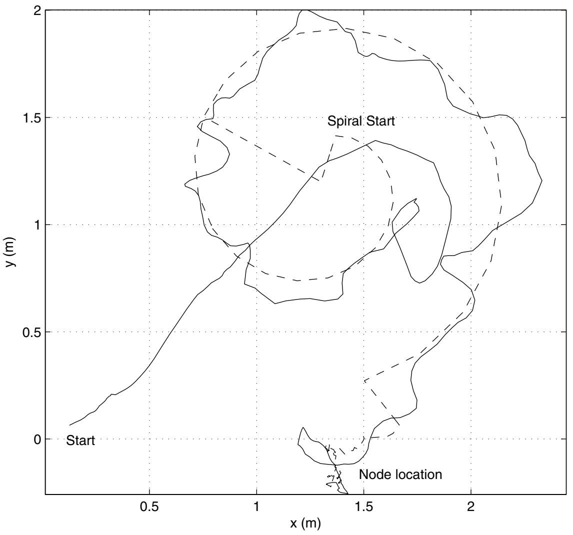


Fig. 5. Spiral search to locate node (solid) actual, (dashed) demand.

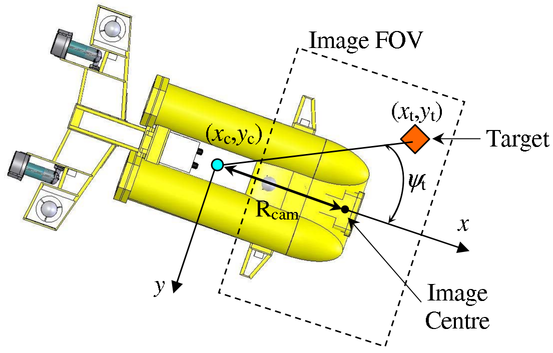


Fig. 6. Coordinate system used for object tracking control.

the station keeping algorithm attempts to keep the node in the image centre. However, if the node is temporarily lost from view, the vehicle servos toward the last known recorded position. If the node fails to respond, or when the data has been downloaded, the vehicle commences motion toward the next node in the tour.

For the vehicle in its full configuration of 6-thrusters the station keeping is a straightforward 2D visual-servo system. Image errors are mapped to surge and sway forces which are converted to velocities by hydrodynamic damping. For these experiments we were missing the tail thruster which is the only one that provides a lateral force, thus making this degree of freedom uncontrollable. Therefore, we require non-holonomic control to maintain station. To determine the vehicle control inputs, it is assumed the target node is at a known coordinate (x_t, y_t) and the vehicle's heading (ψ) is estimated by on-board compass. Further, we have a good estimate of the pivoting radius (R_{cam}) from the camera image centre (x, y) to the vehicles centre-of-gravity (x_c, y_c) as shown in Fig. 6.

With the camera image centre coordinates being a direct output of the visual motion estimation algorithm (x, y), the centre-of gravity coordinates are given by

$$(x_c\ ,\ y_c) = (x - R_{cam}\cos\psi\ ,\ y - R_{cam}\sin\psi) \tag{1}$$

The target's bearing relative to the vehicle's forward axis (ψ_t) can be determined by

$$\psi_t = \tan^{-1}\left(\frac{x_t - x_c}{y_t - y_c}\right) - \psi \tag{2}$$

where $-\pi < \psi_t < \pi$.

Vehicle control is achieved by firstly performing a pure yaw rotation which will turn the vehicle towards the target. However, after rotation there may be some residual x-translation (Δx) required to centre the target in the image. Therefore, Δx is determined from subtracting the pivoting radius from the distance from the centre-of-gravity to the target such that

$$\Delta x = \sqrt{(x_c - x_t)^2 + (y_c - y_t)^2} - R_{cam} \tag{3}$$

As the vehicle can generally move forward much more quickly than it can yaw, it is desired to scale back the forward control input if large yaw angles are required. Therefore, a scaling factor (η_x) is applied to the displacement demand such that

$$\eta_x = \frac{\psi_{max} - \alpha_x}{\psi_{max}} \tag{4}$$

where ψ_{max} is a maximum yaw angle to target at which zero forward demand is desired and $\alpha_x = |\psi_t|$ which is bounded by $(0 \leq \alpha_x \leq \psi_t)$. Therefore, the control inputs for forward velocity (or displacement) and yaw are given respectively by

$$u_{x_{dmd}} = K_x \eta_x \Delta x \tag{5}$$

$$u_{\psi_{dmd}} = K_\psi \psi_t \tag{6}$$

where K_x and K_ψ are control constants of proportionality.

Figure 7 shows the vehicle state for a typical node acquisition and station holding operation. The top two traces show the vehicle's visual odometry-based position estimate, the next shows the identified target area, with the last two traces showing the target location in the image plane. As the communication node attached to the AUV is not co-located with the camera, in this experiment the node was being kept at image coordinates (-0.1,0) to better locate the communication node directly above the target. We can see that the vehicle maintains a good quality lock on the target. At $t = 250$s a significant disturbance was introduced to the vehicle so that it lost the target but quickly reacquired it, which clearly shows the robustness of the controller and vision system.

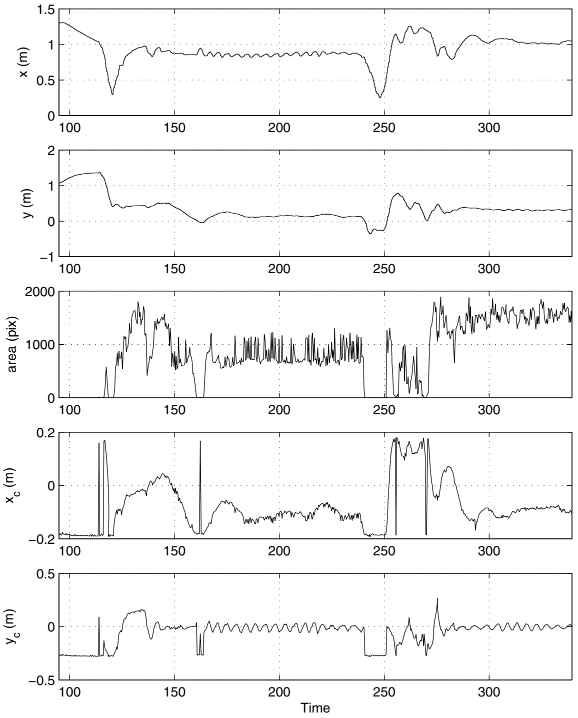


Fig. 7. Vehicle state during visual servoing for typical hover.

For movement between sensor nodes, the same hovering control procedure is applied, however, a pure pursuit strategy is adopted whereby a pseudo target position (x_t, y_t) is incremented at a desired speed toward the desired node position.

E. Data Muling: Communication

Once the Aquafleck is identified, a protocol is executed to establish communication for data transmission. The robot first enquires about the amount of data to upload from the node. The robot then requests data in 239 byte segments. If a segment is not received within a specified period, the robots asks for it again.

Communication is achieved with optical signalling at a data rate of 320 kbits/sec with a stateless protocol and no acknowledgements. The robot can request: the node ID, the node's capabilities (camera, pressure, temperature), the amount of logged data stored in flash memory, and a specific part of the data (specified by address and length). The Aquafleck will respond to all these queries with the information requested.

When data transmission is completed and with the node still in the robot's camera view, the visual odometry is reset to the known location of the current node, and the next node in the sequence is identified.

IV. Experiments

A joint CSIRO and MIT experimental evaluation of the proposed data muling strategy using vision-based node localisation and vehicle control was conducted in the CSIRO AUV test facility. Figure 8 shows the AUV in the pool whilst data muling.

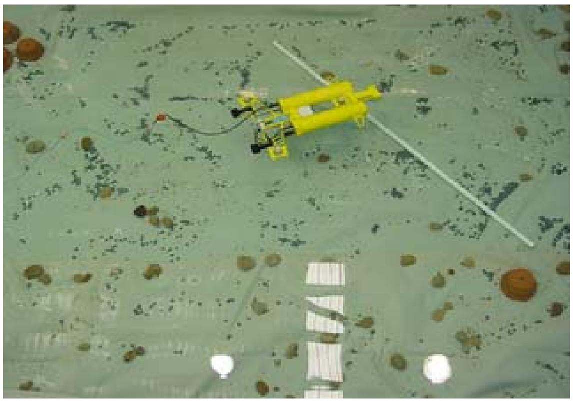


Fig. 8. The Starbug AUV during data muling experiments.

Three sensor nodes were placed in a non-symmetric pattern with each node's approximate Cartesian position recorded as an a priori map. These positions, along with the node identification number, are listed in Table I (note the y direction is aligned with magnetic North).

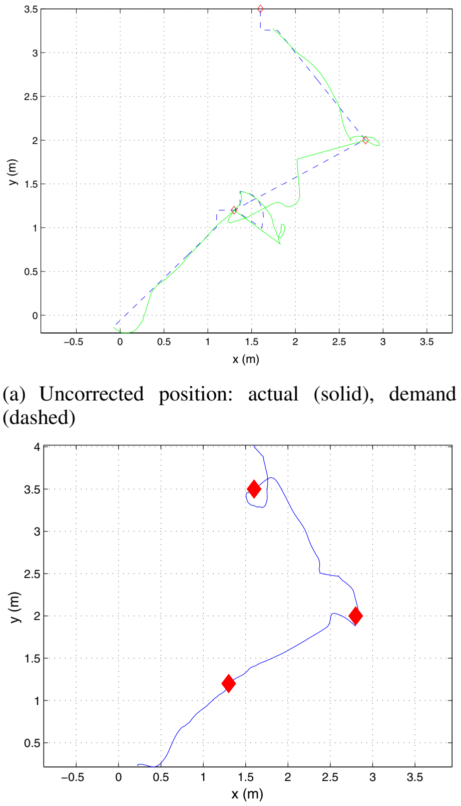
Node ID	x (m)	y (m)
1	1.3	1.2
2	2.8	2.0
3	1.6	3.5

TABLE I

Approximate sensor node location

To verify the performance of the proposed system, a series of experiments were performed where the AUV's starting position was varied throughout the pool. In all experiments, the AUV took its starting location as the origin and using its vision-based position estimate moves in the direction of the map coordinates of Node 1 in an effort to locate the node. If no node is discovered at these coordinates, an increasing spiral search routine is initiated until a node is found. This means that Node 1 is not necessarily the first node discovered and the muling strategy had then to direct the AUV to visit all the other nodes out of order as described in Section III-A.

Figure 9(a) shows the AUV's vision-estimated position during the ideal case of when the starting location is the almost the same as the map origin. The diamonds indicate the approximate location of the nodes from Table I. As can be seen, the vision-based position estimate coincides well with the map-based node position. Once a node is found, the vision system resets its position estimate to that of the node's map coordinates as shown by the position estimate jumps in the vicinity of the node. Figure 9(b) shows the corrected trajectory using the first node's position to back calculate the AUV's initial start position.



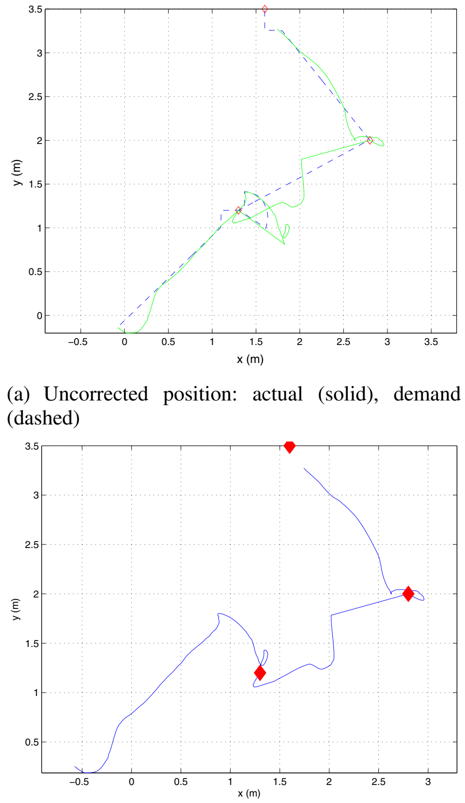
(a) Uncorrected position: actual (solid), demand (dashed)

(b) Corrected AUV position estimate

Fig. 9. Vision-estimated vehicle position and node position with AUV starting location at map origin. Node visitation in ascending order.

Figure 10 shows the results of a second scenario where the AUV's starting position was such that it was guided toward the middle of the node cluster and a search routine had to be initiated. Figure 10(a) shows the vehicle's estimated position for control. The vision-based position "reset" is clearly seen once the node has been found. Figure 10(b) shows the corrected trajectory using the first found node's position to back calculate the AUV's initial start position. In this instance, the nodes are visited in order.

The final scenario consists of the AUV being started in a position which is not the map origin, and the first node found is not Node 1. Figure 11(a) shows the AUV's position estimate during the search when it believes that it started at the map origin and subsequent resetting of the vehicle's position based on the first found node. Figure 11(b) shows the corrected trajectory showing the true vehicle position and actual starting location during the experiment. The spiral search can be clearly seen before locating Node 2.



(a) Uncorrected position: actual (solid), demand (dashed)

(b) Corrected AUV position estimate

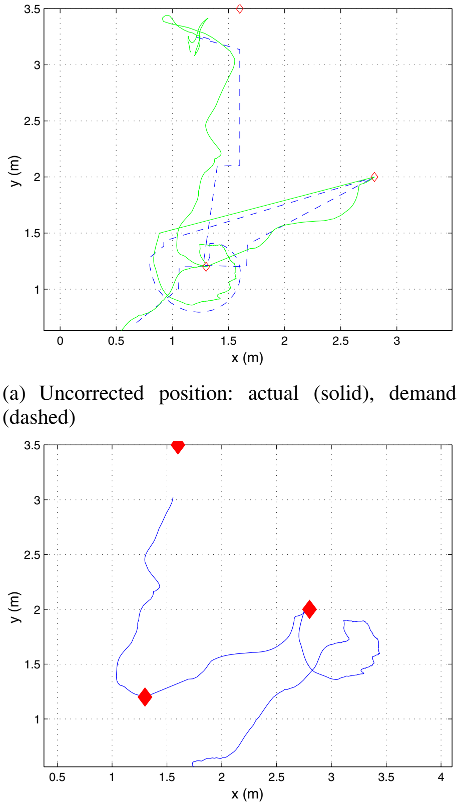
Fig. 10. Vision-estimated vehicle position and node position with AUV starting location offset from map origin. Node visitation in ascending order.

In total, the AUV successfully completed over 46 muling missions locating up to 8 nodes per mission in the test tank. These missions involved visiting the nodes in order, out of order, and through spiral searches to locate the nodes.

V. Conclusions

We have described a system for collecting data from an underwater sensor network using an AUV as a data mule. This is a one example of many useful robot and sensor-network interactions that we are currently studying — others include network deployment, retrieval, post deployment localization, and network repairs or extensions. Such underwater networks can perform many useful tasks such as long-term environmental monitoring or surveillance.

We present algorithms for path planning, navigation and hovering while uploading data. The approach has been tested experimentally with up to eight custom built underwater sensor nodes and an AUV with visual navigation capability. Our future work will integrate an acoustic broadcast capability to signal events that require AUV intervention, and on optimal coverage of the sensor network with multiple AUVs.



(a) Uncorrected position: actual (solid), demand (dashed)

(b) Corrected AUV position estimate

Fig. 11. Vision-estimated vehicle position and node position with AUV starting location offset from map origin. Node visitation out of order.

ACKNOWLEDGMENT

The authors would like to thank the assistance of the CSIRO Autonomous Systems Lab, in particular Les Overs, Stephen Brosnan and Pavan Sikka. We also thank Keith Kotay for his advice on the design and construction of Amour and the Aquaflecks. Support for this work has been provided in part by NSF grants IIS-0426838 and 0225446, ONR PLUSNet program, and Intel. We are grateful for it.

REFERENCES

[1] J. Yuh, “Design and control of autonomous underwater robots: A survey,” *Autonomous Robots*, vol. 8, no. 1, pp. 7–24, 2000.

[2] L. L. Whitcomb, “Underwater robotics: Out of the research laboratory and into the field,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2000, pp. 709–716.

[3] J. J. Leonard, A. A. Bennett, C. M. Smith, and H. J. S. Feder, “Autonomous underwater vehicle navigation,” MIT Marine Robotics Laboratory, Cambridge, MA, USA, Memorandum 98-1, 1998.

[4] D. Wettergreen, C. Gaskett, and A. Zelinsky, “Autonomous control and guidance for an underwater robotic vehicle,” in *Proceedings of the International Conference on Field and Service Robotics*, Pittsburgh, USA, September 1999.

[5] L. L. Whitcomb, D. R. Yoerger, and H. Singh, “Combined Doppler/LBL based navigation of underwater vehicles,” in *Proceedings of the 11th International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, USA, August 1999.

[6] S. Cowen, S. Briest, and J. Dombrowski, “Underwater docking of autonomous undersea vehicles using optical terminal guidance,” in *Oceans97, MTS/IEEE Conference*, 1997.

[7] J. G. Bellingham, C. A. Goudey, T. R. Consi, J. W. Bales, D. K. Atwood, J. J. Leonard, and C. Chryssostomidis, “A second generation survey AUV,” in *IEEE Conference on Autonomous Underwater Vehicles*, Cambridge, MA, USA, 1994.

[8] H. Singh, M. Bowen, F. Hover, P. LeBas, and D. Yoerger, “Intelligent Docking for an Autonomous Ocean Sampling Network,” in *Oceans97, MTS/IEEE Conference*, Halifax, October 1997.

[9] S. Kalantar and U. R. Zimmer, “Contour shaped formation control for autonomous underwater vehicles using canonical shape descriptors and deformable models,” in *Proceedings of the IEEE International Conference on Marine Technology and Ocean Science*, Kobe, Japan, 2004.

[10] C. C. Eriksen, T. J. Osse, R. D. Light, T. Wen, T. W. Lehman, and P. L. Sabin, “Seaglider: A long-range autonomous underwater vehicle for oceanographic research,” *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 424–436, October 2001.

[11] E. Fiorelli, N. E. Leonard, P. Bhatta, D. Paley, R. Bachmayer, and D. M. Fratantoni, “Multi-AUV control and adaptive sampling in Monterey Bay,” in *Proceedings of the IEEE Autonomous Underwater Vehicles: Workshop on Multiple AUV Operations*, Sebasco, ME, USA, June 2004.

[12] “Autonomous Ocean Sampling Network (AOSN)II, collaborative project,” http://www.princeton.edu/d̃csl/aosn/.

[13] V. Bokser, C. Oberg, G. S. Sukhatme, and A. A. Requicha, “A small submarine robot for experiments in underwater sensor networks,” in *International Federation of Automatic Control Symposium on Intelligent Autonomous Vehicles*, 2004.

[14] K. McIsaac and J. Ostrowski, “A geometric approach to anguilliform locomotion: Modelling of an underwater eel robot,” in *Proceedings of the IEEE Conference of Robotics and Automation*, 1999, pp. 2843–2848.

[15] J. Ayers, C. Wilbur, and C. Olcott, “Lamprey robots,” in *Proceedings of the International Symposium on Aqua Biomechanisms*, 2000.

[16] P. Sikka, P. Corke, and L. Overs, “Wireless sensor devices for animal tracking and control,” in *Proc. First IEEE Workshop on Embedded Networked Sensors*, Tampa, Florida, Nov. 2004.

[17] M. Dunbabin, J. Roberts, K. Usher, G. Winstanley, and P. Corke, “A hybrid AUV design for shallow water reef navigation,” in *Proceedings of the 2005 International Conference on Robotics and Automation*, Barcelona, Apr. 2005, pp. 2117–2122.

[18] M. Dunbabin, K. Usher, and P. Corke, “Visual motion estimation for an autonomous underwater reef monitoring robot,” in *Proc. International Conference on Field & Service Robotics*, Port Douglas, 2005, pp. 57–68.