

Unsupervised Learning of Terrain Appearance for Automated Coral Reef Exploration

Philippe Giguere, Gregory Dudek, Christopher Prahacs, Nicolas Plamondon
Centre for Intelligent Machines
McGill University
Montreal, Quebec, Canada H3A 2A7
Email: {philg,dudek,cprahacs,nicola}@cim.mcgill.ca

Katrine Turgeon
Dept. of Biology
McGill University
Montreal, Quebec, Canada H3A 2A7
Email: katrine.turgeon@mail.mcgill.ca

Abstract

We describe a navigation and coverage system based on unsupervised learning driven by visual input. Our objective is to allow a robot to remain continuously moving above a terrain of interest using visual feedback to avoid leaving this region. As a particular application domain, we are interested in doing this in open water, but the approach makes few domain-specific assumptions. Specifically, our system employed an unsupervised learning technique to train a k -Nearest Neighbor classifier to distinguish between images of different terrain types through image segmentation. A simple random exploration strategy was used with this classifier to allow the robot to collect data while remaining confined above a coral reef, without the need to maintain pose estimates. We tested the technique in simulation, and a live deployment was conducted in open water. During the latter, the robot successfully navigated autonomously above a coral reef during a 20 minutes period.

1. Introduction

Underwater marine environments provides for challenging problems relevant to the robotics community. A significant constraint from these environment is the electrical conductivity of salt water, preventing the propagation of radio waves. This limits the bandwidth and range available for communication with untethered vehicles. Furthermore, GPS signals can not be used for localization. Given these constraints, the level of autonomy required to accomplish

certain tasks such as surveying or mapping is significantly increased, compared to land vehicles.

Our paper addresses the automated collection of visual survey data based upon visual input. This process is accomplished in three steps: in the first step, a training set of unlabeled images is collected. In the second step, a novel clustering algorithm is employed to train a classifier to recognize the different terrain types. Finally, a simple exploration strategy employing this trained classifier is used to modify the direction of the robot.

Our experiments dealt with the specific case of visually mapping a small coral head using unsupervised learning of the appearance of the coral compared to its surroundings. We used reefs for our experiments since we work with biologists who have a particular interest in such domains, and who can provide us with accurate ground truth data for testing purposes.

In this paper we will discuss the algorithmic approach we used. We then consider its behavior using manually collected data and a motion simulator. We do this since during open water experiments, ground truth measurements are notoriously unreliable. In order to keep the visual observations coherent in open water, we employed a controller to maintain a fixed depth as the robot navigated above the reef. Finally, we present the results from the full system in open water experiments conducted over a reef using a robot we have previously developed.

2. Previous Work

2.1. Autonomous Underwater Surveying

A number of teams have worked towards automating the process of underwater surveys. A key component of those systems is the use of a pose estimator. For example, in Eustice *et al.* [1], an efficient SLAM technique combined SIFT-based visual landmark and inertial information in a Extended Information Filter (EIF) to map the wreck of the Titanic. In Dunbabin *et al.* [2], ego-motion was estimated using 3D image registration and weak inertial cues. In Corke *et al.* [3], two methods of underwater localization systems were compared: one relied on acoustic signals from known underwater nodes, and the other on the integration of scaled optical flow. A large scale (48 km²) sonar survey was accomplished in Grasmueck *et al.* [4], using a combination of inertial and acoustic localization. These techniques offer precise localization over a reef, an essential component of mapping task. On the other hand, a known path covering the area to be explored must be pre-programmed into the system by a human operator.

2.2. Image-Segmentation

Image segmentation is a process by which a digital image is segmented into different regions, based on a set features. Clustering methods such as K-means have been employed for this purpose in Chen *et al.* [5]. Region growing methods such as Zhu *et al.* [6] have been employed as well. The closest technique to what we are presenting here is called the Normalized-cut or *N-cut* segmentation algorithm, introduced by Shi *et al.* [7].

3. Unsupervised Learning of Terrain Appearance through Image Segmentation

3.1. Extension of a Previous Clustering Algorithm for Image Segmentation

A key result in this paper is the extension of a previous clustering algorithm described in Giguere *et al.*[8]. This algorithm performs clustering of time-series by exploiting continuity in the temporal dimension. It uses a classifier to compute probability estimates. Clustering is achieved by training this classifier to minimize an objective function. The continuity in the temporal dimension is preserved by penalizing rapid variations of probability estimates over time. A normalizing term is also used to avoid solutions that are not discriminative, such as solutions where all the

samples are grouped in the same class. The training rule is:

$$\arg \min_{\vec{\theta}} \sum_{c=1}^{|C|} \frac{\sum_{t=1}^{T-1} (p(o_c|\vec{x}_{t+1}, \vec{\theta}) - p(o_c|\vec{x}_t, \vec{\theta}))^2}{\text{var}(p(o_c|\vec{X}, \vec{\theta}))^2} \quad (1)$$

where $\vec{\theta}$ are the classifier parameter values, and $p(o_c|\vec{x}, \vec{\theta})$ is the posterior probability of samples $x \in X$ being class $c_i \in C$ given the classifier parameters $\vec{\theta}$. x is assumed to be a set of representative features.

In image segmentation, continuity is preserved between adjacent pixels or regions of an image. The same condition can be incorporated in Eq. 1. If the original image is subdivided into $n \{I_1, \dots, I_n\}$ non-overlapping sub-images, and we assume a four neighbor connectivity in the image, the numerator of Eq. 1 becomes:

$$\Delta_p(c) = \sum_{i=1}^{n-1} \{w_x(i)(p(o_c|f(I_{s_d(i)}), \vec{\theta}) - p(o_c|f(I_i), \vec{\theta}))^2 + w_y(i)(p(o_c|f(I_{s_r(i)}), \vec{\theta}) - p(o_c|f(I_i), \vec{\theta}))^2\} \quad (2)$$

where $s_r(i)$ is the index of the sub-image to the immediate right of i , $s_d(i)$ is the index of the sub-image below i , and $f(I_i)$ is a feature-extraction function computed over the image I_i . The differences are weighted by $w_x(i)$ and $w_y(i)$, where:

$$w_x(i) = \begin{cases} 0 & \text{if sub-image } i \text{ is on the right edge} \\ 1 & \text{otherwise} \end{cases}$$

and

$$w_y(i) = \begin{cases} 0 & \text{if sub-image } i \text{ is at the bottom edge} \\ 1 & \text{otherwise} \end{cases}$$

This effectively removes the continuity preservation at the image boundary. The training rule becomes:

$$\arg \min_{\vec{\theta}} \sum_{c=1}^{|C|} \frac{\Delta_p(c)}{\text{var}(p(o_c|\vec{X}, \vec{\theta}))^2} \quad (3)$$

Eq. 2 demonstrates that the two dimensional problem can be solved as the summation of two, one-dimensional problems. The first one dimensional problem is represented by a sequence of probability estimates taken from sub-images in left-to-right, top-to-bottom order. The second sequence is created by taking the probability estimates in top-to-bottom, left-to-right order. Fig. 1 shows this ordering for an image divided in 9 sub-images.

This is in contrast with a Markov random field (MRF), where the 2D problem cannot be transformed into a 1D problem. Recalling that for a MRF, the probability estimates for a given location depend on the estimates of a set of neighboring locations. In our case, this dependency is taken care by the process of the cost minimization; the probability estimates themselves are independent of the neighboring location's estimates.

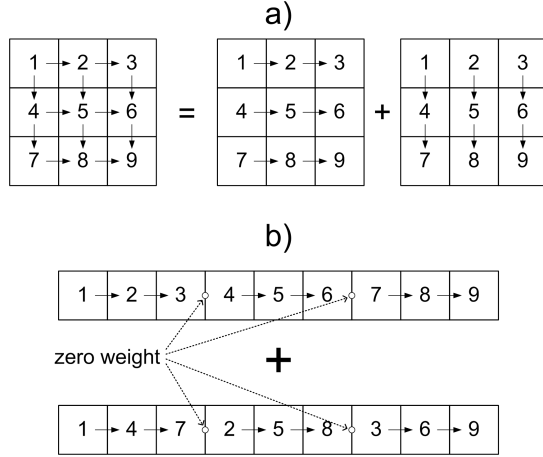


Figure 1. a) An image is divided into 3x3 sub-images. Each square represents a probability estimate computed from a classifier, using extracted features. Arrows indicate neighboring differences used in the objective function to preserve image continuity. b) The problem can be re-framed into a 1-D continuity preservation, as shown in b).

3.2. k-Nearest Neighbor Classifier

A k -Nearest Neighbor classifier is an instance-based classifier used in pattern recognition. A sample x is classified by comparing it to its k -nearest neighbor in a labelled set \vec{X} . The distance metric used in feature space is generally the Euclidean distance, although others can be used. One advantage of using such classifier is that there is no assumptions made about the shape of the distributions, contrary to other approaches such as mixture of Gaussians classifier where the distributions are assumed to be normal.

The probability that a given sample x belongs to class c is computed as:

$$p(c|x, \vec{\theta}) = \frac{1}{k} \sum_{i=1}^k h_{ck} \quad (4)$$

where $\{h_{c1}, \dots, h_{ck}\} \in \vec{\theta}$ are the k -closest samples in feature space to x , and $0 \leq h_{ck} \leq 1$ is the probability that sample k belongs to class c .

It has been shown that this classification method has at most twice the Bayes probability of error [9] using $k = 1$ neighbor, and that the probability of error asymptotically approaches the Bayes error [10], as k and the number of samples in the training set goes to infinity.

3.3. Using the k -Nearest Neighbor Classifier

As mentioned in Section 3.1, our clustering algorithm works by finding the classifier parameter values by minimizing a specific cost function. In this paper, we employed a k -nearest neighbor classifier. A significant conceptual difference exists between the normal use of the k -nearest neighbor classifier and our use. Normally, the set X is labelled and the classifier is strictly concerned about labelling sample q outside of X . In our case, the set X is unlabeled, and the probabilities are computed for $x \in X$ using the same set X .

3.4. Image Segmentation on Coral Reef Picture

Given images containing a set of terrains which are visually different, it is possible to learn their individual appearance by performing image segmentation [11]. This image should contain a limited set of terrains, and they must be different in appearance as well. Images of coral reefs generally shows these properties, their surrounding environment being covered with sand. Therefore, image segmentation technique should perform well for this type of environment.

We tested the image segmentation algorithm on a pre-existing 6,500 by 6,500 pixels image of a coral reef, shown in Fig. 2 a). This image was the results of mosaicking pictures taken manually with an underwater camera by a team of biologists (K. Turgeon, unpublished data). The pictured reef had a sufficiently large band of sand surrounding it to isolate it from other reefs. The sand-covered area was manually expanded by copy-paste, since it was incomplete in the original image. We used the following four features in the image segmentation and classification processes: hue, saturation and value from the HSV representation of the image, and the output of a band-pass filter having lower and upper cutoff frequencies of 3 and 20 cycles per sub-window, respectively.

Training a classifier this way is a key component of the exploration strategy described in Section 4. This training algorithm provided the robot with the capability to autonomously identify the surface type below.

4. Autonomous Reef Exploration Strategy

We developed an exploration strategy for an underwater robot to confine its motion above a coral reef. This strategy was adapted from earlier work on surface texture following for a blind differential-drive robot in Giguere *et al.* [12]. It was assumed that a robot using this strategy was capable of executing two simple locomotion commands in a stable plane: forward motion, and yawing at a rate of $\dot{\theta}$.

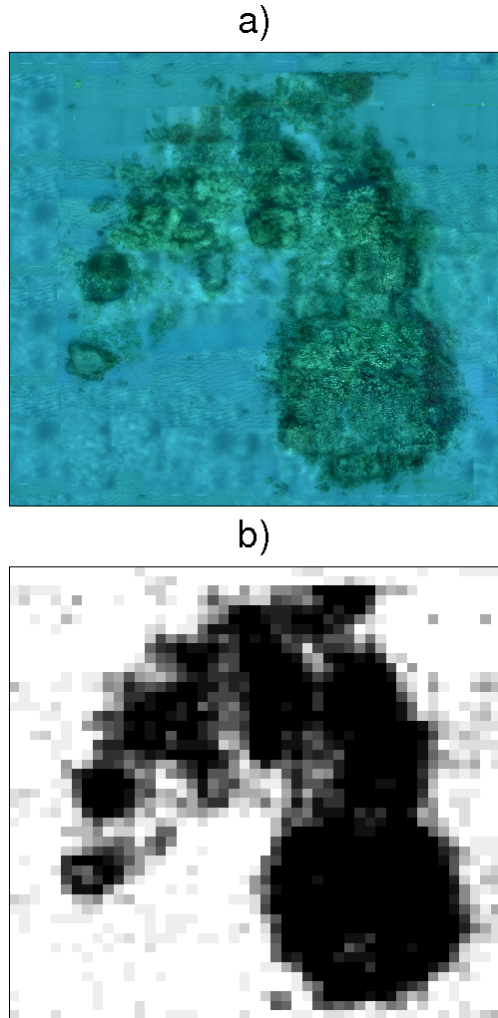


Figure 2. 6,500 x 6,500 *pixels* image of a coral reef in a). Probability estimates for 100x100 *pixels* sub-images computed with the *k*-Nearest Neighbor classifier trained using Eq. 3 in b). Darker regions are associated with the presence of coral.

The primary goal of this strategy was not to provide good coverage of the reef, but rather to demonstrate the practicality of our unsupervised learning algorithm in the context of autonomous robotics. This strategy can be loosely defined as being a form of visual servoing, since it relied primarily on vision to control the position of the robot.

4.1. Phase 1: Collecting the Training Data Set

A single large image of the coral reef and its environment is generally not available before a trial to perform im-

age segmentation. Instead, the segmentation process had to be done over a collection of partial images of the reef and its surrounding environment. These images were collected by an operator or the robot, and were not labeled. Some care had to be taken to ensure that both environment (sand and coral in our case) were represented in comparable amount in this training set. If not, the over-representation of one terrain might force the training algorithm to include part of it into the other terrains. This in turn will result in decreased classification performances.

4.2. Phase 2: Training the Visual Classifier

The segmentation described in Section 3.4 was performed over a meta-image. It was built by concatenating the images of the training set. The weight vectors $w_x(i)$ and $w_y(i)$ of Eq. 2 associated with the boundaries between the images were set to zero, thus removing the continuity preservation for those locations. The result of the segmentation was a trained classifier that was used to identify the terrain.

4.3. Phase 3: Exploring the Reef

At each iteration, the downward looking camera took snapshots of the surface below. A state machine controlled the motion of the robot, and transitions between states were triggered by changes in the image classification. The state machine had three states:

- *GoStraight*
- *SeekCoral*
- *CounterTurn*

The *GoStraight* state sent a constant forward velocity command to the robot. The state machine transitioned from *GoStraight* to *SeekCoral* when the classification of the camera image went from coral to sand.

The *SeekCoral* state sent a yawing command to the robot. The yawing rate command decreased over time, generating a spiral-shaped trajectory. Once an image of the downward looking camera was classified as coral, the state machine transitioned from *SeekCoral* to *CounterTurn*.

During the *CounterTurn* state, a yawing command in the opposite direction was sent to the robot. This command lasted for a random duration, emulating a random-walk type strategy of exploration. The state went back to *GoStraight* once this yawing command was completed. Fig. 3 presents these steps for a complete cycle of the state machine.

The yawing directions of the *SeekCoral* state was reversed for every ten cycles of the state machine. This was done to avoid tangling a fiber optic tether, if attached to the robot. It also provided better coverage of the reef.

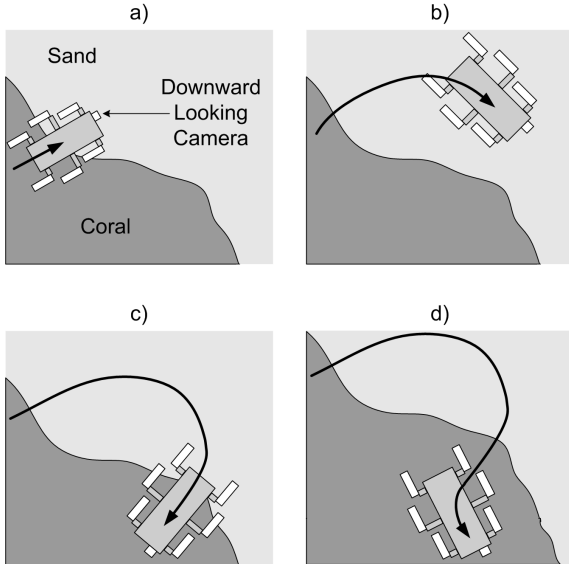


Figure 3. Strategy used to confine the robot above a coral reef. a) The robot, while in *GoStraight* state, classified the image of the downward looking camera as sand. b) The robot transitioned to the *SeekCoral* state, and turns (yaws) in the right-hand direction, seeking coral. c) The coral reef is detected. d) The robot changed state to *CounterTurn*, and turns (yaws) in the opposite direction for a small random duration, heading towards a different region of the reef.

5. Simulation

5.1. Description

We used a simulation to assess the viability of the strategy. The simulation computed the motion of the robot and generated an appropriate image for the camera. The simulated robot dynamics were simplified by considering it massless. The robot accepted two motion commands: forward motion and yawing rotation command $\dot{\theta}$. The simulation restricted the motion in a 2D plane, corresponding to a fixed depth above a submerged coral reef. It did not take into account hydrodynamic forces or water currents, both of which require computing-intensive fluid dynamics simulation.

The image scale was set to 40 *pixels/m*. The simulated camera image was generated by selecting the area of the complete coral reef image corresponding to the simulated x and y locations of the robot, and rotated to match the robot orientation θ .

5.2. Results

A k -Nearest Neighbor classifier was trained using the technique described in Section 3.4 with 40 unlabeled sub-images selected from the original image shown in Fig. 2 a). About 40 *percent* of the images contained either sand or coral, and 20 *percent* contained a mixture of both. This emulated phase one in Section 4. The exploration strategy was then run for a number of iterations. The whole process was repeated ten times, with success. Fig. 4 shows one particular run of the simulation, with 4000 iterations. As can be seen from the figure, the robot successfully covered most of the reef, and managed to return to the reef every time.

6. Coral Reef Field Trials

6.1. Description of the Robot

The vehicle used in the experiments, *Ramius*, (seen in Fig. 5) is a hexapod robot specifically designed for amphibious locomotion. It is the direct descendant of an earlier prototype called Aqua [13], itself adapted from the successful RHex platform [14]. The six motors are fitted with flippers, providing thrust underwater in five degrees of freedom. Two PC/104 single-board computers, one a 300 MHz Pentium-equivalent running QNX and the other a 1400 MHz Pentium-M running GNU/Linux, are used for on-board computation. Communication to a remote operator laptop is done over Firewire transmitted via a fiber optic tether. This family of robots has been used extensively in field trials, notably in visual-servoing tasks described in Sattar *et al.* [15],[16].



Figure 5. Picture of the swimming robot *Ramius* while being deployed in Barbados.

A proportional-derivative linear controller [17] in the Stability Augmentation System (SAS) maintained the pitch



Figure 4. Path of the robot taken during one simulation run of 4000 iterations, overlaid over the coral reef image used to generate the simulated camera frames.

and roll angles of the robot. The SAS modified the swimming pattern of the flippers to generate proper pitch or roll correcting moments. The yaw angle was left uncontrolled, so the coral reef exploration strategy presented in Section 4 could modify the heading of the vehicle. A simple autopilot maintained the depth of the robot at a constant value of 4 ± 0.2 m, using small pitching corrections executed by the SAS.

6.2. Experimental Setup

We evaluated the performance of this algorithm and approach over a coral reef in Barbados. The experiment involved deploying the robot using a team comprising an operator, divers underwater to assess the vehicle’s performance, and support staff. The selected coral reef had a rough circular shape, with a diameter of approximately 5 m. A laptop on the surface, connected via a fiber optic tether, provided the user interface to the robot and allowed its behavior to be monitored and adjusted. While the robot’s on-board systems controlled many aspects of the robot’s behavior, the particular algorithm described in this paper in Section 4 (including the image processing) was executed off-board on the laptop. Vehicle control commands were

then returned via the fibre-optic link.

Training data for the controller was collected by having a scuba diver manually move the robot in the region of interest. This region contained both sand and coral. During this training interval, the downward looking camera on the robot captured a training sequence of 10 images, at a rate of one image every 5 seconds. Note that the diver was not aware when images were being taken, so that it could not influence their choice. The training algorithm described in Section 3.4 used this unlabeled sequence of images from the different regions and generated a trained k -Nearest Neighbor classifier.

Fig. 7 shows one such training image sequence and the classification results. Following the training, the diver placed the robot above the coral reef, and the exploration algorithm in Section 4 was enabled by a human operator at the surface.

6.3. Trial Results

Several failed attempts were executed before the system was finally able to perform as intended. First, the initial turning radius had to be increased to alleviate the impact of sideslip, a lateral sliding motion, during turning maneuvers.

Without the increase, the robot motion during rotation was such that it would rotate around an axis close to the center of the downward-looking camera field of view. This resulted in capturing the same but rotated image, and the robot had difficulties finding the coral reef again. Another issue had to do with the captured training sequence, which often did not contain enough images from a particular surface, or contained part of the diver. This confused the training algorithm in Section 3.4, resulting in poor classification of images and poor performance of the exploration strategy.

In the single successful trial, the robot managed to swim autonomously above the coral reef for a period of 20 *minutes*. A video sequence of the experiment was recorded, and a still frame is shown in Fig. 6. A sequence of states recorded during the trial is shown in Fig. 8 a), along with the sent yaw command in Fig. 8 b).

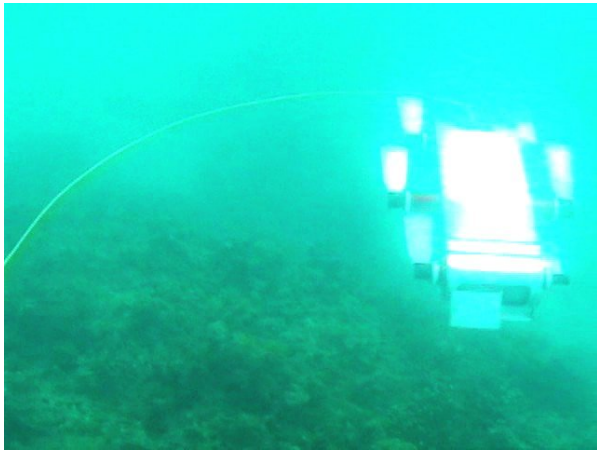


Figure 6. Still frame extracted from video taken during the experiment. Coral reef is the the darker area at the bottom. Significant presence of silt in the water diffused the light, blurring images during the experiment.

7. Conclusion and Future Work

In this paper, we presented an novel extension to a clustering algorithm allowing image segmentation. We successfully applied this technique to images of coral reefs and sand, obtaining classifiers that segmented the image along terrain types. We devised a simple random exploration strategy that employed this classifier. This enabled autonomous coral reef exploration requiring minimal human intervention. Simulation results confirmed the viability of the strategy, and field trials over a live coral reef in open waters yielded one successful autonomous run lasting over 20 *minutes*. We feel confident that more successful trials will be achieved in the next round of experiments. This stems from

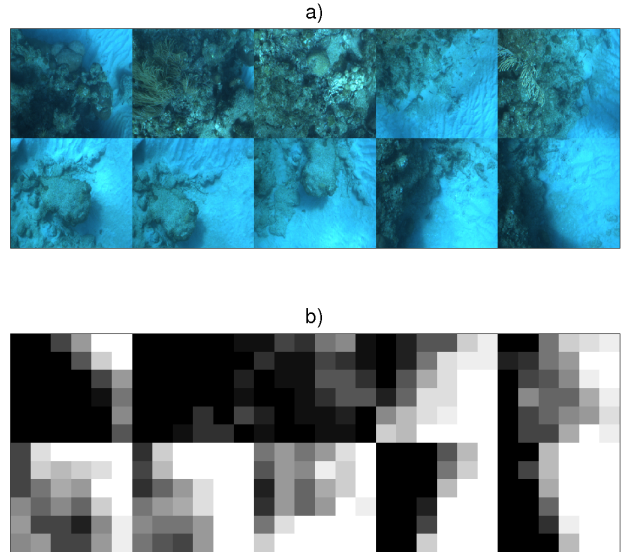


Figure 7. Meta-image built from the training data set used during the successful trial in Barbados in a). Classifier probability estimates (shown as grey-scale) after training, for corresponding sub-images in b).

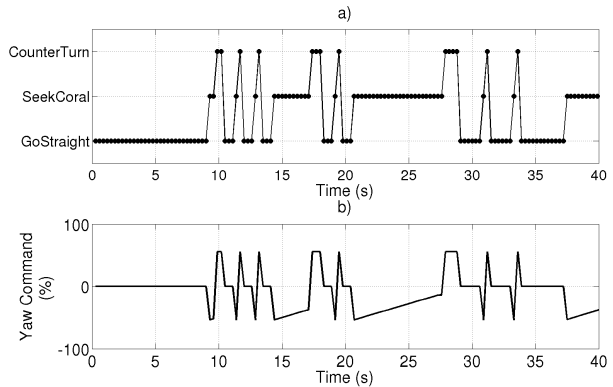


Figure 8. State sequence in a) during part of the successful trial. Yaw commands sent to the robot, in b). A full 100 percent command translates into a yawing rate of approximately 30 $^{\circ}/sec$.

the exceptionally good behavior of the system during the successful trial.

The experience gained in the field trials pointed to certain deficiencies. In particular, we plan to increase the robustness of the detection by employing better visual features. We are also looking at reducing the time needed for the robot to find coral, by including weak inertial cues to add about-turns behaviors in the system.

During the trials, the robot was connected to an external laptop via a fiber optic tether. While this was not generally necessary to operate the vehicle, it allowed us to make final adjustments to the exploration strategy. In view of these successful results, we are planning to fully embed all aspects of the control software in the on-board systems. This will make running these experiments much easier, and provide in the future for an intelligent autonomous platform ready to be deployed.

Finally, by also capturing stereo images and inertial information during the exploration, we are interested in augmenting our representation with a full 3-D model (computed off-line).

Acknowledgment

The authors would like to thank Yiannis Rekleitis for the picture in Fig. 5 and, his suggestions and his help during the experiments, and Anna Topol for her help during the trials.

References

- [1] Ryan Eustice, Hanumant Singh, John Leonard, Matthew Walter, and Robert Ballard. Visually navigating the rms titanic with slam information filters. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.
- [2] Matthew Dunbabin, Kane Usher, and Peter Corke. Visual motion estimation for an autonomous underwater reef monitoring robot. In Peter I. Corke and Salah Sukkarieh, editors, *FSR*, volume 25 of *Springer Tracts in Advanced Robotics*, pages 31–42. Springer, 2005.
- [3] P. Corke, C. Detweiler, M. Dunbabin, M. Hamilton, D. Rus, and I. Vasilescu. Experiments with underwater robot localization and tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4556–4561, April 2007.
- [4] Mark Grasmueck, Gregor P. Eberli, David A. Viggiano, and Thiago Correa. Autonomous underwater vehicle (auv) mapping reveals coral mound distribution, morphology, and oceanography in deep water of the straits of florida. 33, 2006.
- [5] C.W. Chen, J.B. Luo, and K.J. Parker. Image segmentation via adaptive k-mean clustering and knowledge-based morphological operations with biomedical applications. 7(12):1673–1683, December 1998.
- [6] Song Chun Zhu and Alan Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, 1996.
- [7] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.
- [8] Philippe Giguere and Gregory Dudek. Clustering sensor data for terrain identification using a windowless algorithm. In *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland, June 2008.
- [9] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.
- [10] Luc Devroye, Laszlo Györfi, and Gabor Lugosi. *A Probabilistic Theory of Pattern Recognition (Stochastic Modelling and Applied Probability)*. Springer, February 1997.
- [11] Charles W. Therrien. An estimation-theoretic approach to terrain image segmentation. *Computer Vision, Graphics, and Image Processing*, 22(3):313–326, 1983.
- [12] Philippe Giguere and Gregory Dudek. Surface identification using simple contact dynamics for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2009.
- [13] Gregory Dudek, Michael Jenkin, Chris Prahacs, Andrew Hogue, Junaed Sattar, Philippe Giguère, Andrew German, Hui Liu, Shane Saunderson, Arlene Ripsman, Saul Simhon, Luiz Abril Torres-Mendez, Evangelos Miliotis, Pifu Zhang, and Ioannis Rekleitis. A visually guided swimming robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, August 2005.
- [14] Martin Buehler Uluc Saranlı and D. E. Koditschek. *RHex: A Simple and Highly Mobile Hexapod Robot*, 20(1):616 – 631, July 2001.
- [15] Junaed Sattar, Philippe Giguère, Gregory Dudek, and Chris Prahacs. A visual servoing system for an aquatic swimming robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, August 2005.
- [16] Junaed Sattar and Gregory Dudek. A boosting approach to visual servo-control of an underwater robot. In *Proceedings of the 11th International Symposium on Experimental Robotics, ISER*, Athens, Greece, July 2008.
- [17] Nicolas Plamondon and Meyer Nahon. Trajectory tracking controller for an underwater hexapod vehicle. In *Oceans 08 MTS/IEEE*, Quebec City, Canada, September 2008.