2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

MARE: Marine Autonomous Robotic Explorer

Yogesh Girdhar, Anqi Xu, Bir Bikram Dey, Malika Meghjani,
Florian Shkurti, Ioannis Rekleitis, and Gregory Dudek

***Abstract*— We present MARE, an autonomous airboat robot that is suitable for exploration-oriented tasks, such as inspection of coral reefs and shallow seabeds. The combination of this platform's particular mechanical properties and its powerful software framework enables it to function in a multitude of potential capacities, including autonomous surveillance, mapping, and search operations. In this paper we describe two different exploration strategies and their implementation using the MARE platform. First, we discuss the application of an efficient coverage algorithm, for the purpose of achieving systematic exploration of a known and bounded environment. Second, we present an exploration strategy driven by surprise, which steers the robot on a path that might lead to potentially surprising observations.**

Keywords — exploration, coverage, surprise, autonomous surface craft, robot programming.

I. Introduction

Exploring marine environments is a challenging task for the robotics community. We address this challenge by introducing MARE: Marine Autonomous Robotic Explorer. MARE is a sea-worthy and collapsible airboat robot that can carry out autonomous visual exploration of a given region over an ocean or a lake. It is a differential-drive, air-propelled vehicle equipped with a downward-pointing camera, as shown in Fig. 1. Its chassis follows a catamaran hull design, which makes it hydro-dynamically stable and also allows it to accommodate heavy and voluminous payloads. Its air propellers enable MARE to explore marine environments with minimal disturbance to the water surface and the aquatic life. These factors together make MARE a favorable choice as a sea-worthy surface exploration platform, with numerous application possibilities.

In this paper we discuss two different kinds of exploration strategies. These techniques are described in the specific context of the MARE platform within a marine environment, although the algorithmic bases are general enough to be deployed on a variety of different robotic platforms.

Given a map that indicates areas to be explored, we present an efficient coverage system [1] for non-holonomic vehicles that can produce a motion path that sweeps through the entirety of the free space while ensuring minimal overlap. While following this path, the robot takes images using its downward-looking camera and employs an online summarization strategy [2] to detect salient frames that are sufficiently different from previous observations. These key images are stored within a dynamically updated image set, which serves as a summary of the observed scenes during the coverage-based exploration session.

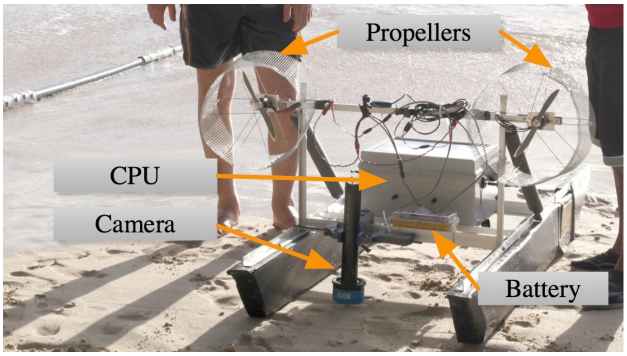


Fig. 1. MARE is an autonomous airboat made from off-the-shelf components, and is capable of exploring turbulent open water environments. Its discreet air-propelled design and hydro-dynamically stable catamaran hull structure make MARE suitable for long-term deployment in all types of bodies of water. Since it has no moving parts in contact with water, MARE can explore marine ecosystems while introducing minimal disturbance.

In situations where we do not have a map of the region, or if the environment to be explored is dynamically changing, then a surprise-driven exploration strategy becomes more suitable. Given a measure of surprise, the exploration task can be formulated as choosing a motion heading that leads to the observation of most surprising events. Since it is impossible to compute the optimal solution to this problem in an online setting, we instead propose a greedy online exploration strategy.

To manage the execution of these exploration strategies while automatically taking into account constraints such as range of operation, wireless connectivity to home base, and battery levels, we propose the use of a powerful software framework and accompanied programming tool called Graphical State Space Programming (GSSP) [3]. GSSP allows the robot operator to program execution plans for experiment sessions, where each plan can comprise of a set of location-specific activities, different reactive behaviors, as well as various failsafe mechanisms. The back-end counterpart to this programming tool then regulates the prioritized execution of these plan components in a structured manner during field operations.

II. Related Work

The development of Autonomous Surface Crafts (ASC), such as our MARE platform, have been mostly restricted to research domains, as opposed to the wide-spread commer-

The authors are with the Centre for Intelligent Machines, McGill University, Montréal, Canada. {yogesh, anqixu, bikram, malika, florian, yiannis, dudek}@cim.mcgill.ca

The authors wish to thank Junaed Sattar, Chatavut Viriyasuthee, Jimmy Li, Simon Norsworthy, and all members of the Mobile Robotics Lab for their contributions to the MARE project.

cialization of Autonomous Underwater Vehicles (AUV) [4]. This trend however has been changing recently with the increasing scope of applications that are possible due to the collaboration of ASCs with AUVs. ASCs can provide a local medium for long-range communication and guidance support for navigation to the AUVs, since radio frequency transmissions required for global referencing and communication are significantly attenuated underwater [5]. GPS information for underwater vehicles is essential for various oceanic applications such as fisheries stock assessments, marine archeology, ecosystem monitoring, and habitat characterization; a compilation of these applications can be found in [6]. A notable system for cooperative localization of an AUV using ASCs was achieved through the use of acoustic modems [7]. Our long-term research objectives share the desire of using the MARE platform to establish a communication network between autonomous underwater vehicles as well as unmanned aerial vehicles (UAV).

Surface crafts can also be used independently, either in groups or individually, for applications such as surface water sampling, hydrographic surveying, bathymetric mapping, and harbor patrolling [5]. Benjamin *et al.* [8] addressed several critical concerns when operating multiple ASCs, including traffic management and obstacle avoidance.

Zhang and Sukhatme [9] designed a system where an ASC interacts with a static sensor network for the purpose of aquatic observation. The sensor network comprised of static nodes that were spread across a region of interest and established spatial coverage of the environment, while an ASC navigating through the network complimented with temporal coverage updates. The static nodes also provided path guidance for the ASC to efficiently navigate through and perform coverage of a given region. One shortcoming of this approach however is that the static nodes need to be distributed based on prior knowledge about locations of interest, and thus do not account for any new interesting events that may occur during the sampling phase. In contrast, our surprise-driven exploration strategy enables our MARE platform to autonomously survey a given region based on the dynamic interestingness of local events.

In another application [10], a modified airboat robot was used to collect surface water samples at various designated points of interest. An extended Kalman filter was used for localization of the ASC, in order to compute desired headings towards the given waypoints.

Finally, a detailed study on the design developments of ASCs over the last 15 years is provided in [4].

III. MARE Platform

A. Hardware Design

The $1.6\,m \times 0.6\,m \times 0.6\,m$ body of our MARE platform is based on an open catamaran hull design and is sufficiently stable to operate within turbulent open water environments. The collapsible wire-frame chassis design also facilitates transportation, deployment, and maintenance. Propulsion is achieved using two air propellers in a differential drive configuration. This is favorable compared to conventional rudder-based propulsion because our vehicle has no moving parts in the water, which minimizes concerns of structural damage due to corrosion, entanglement with marine plants, and general disturbance to the environment. In addition, differential drive can potentially allow in-place rotations, although our robot currently behaves non-holonomically due to hardware restrictions; these will be addressed in a future hardware revision. The motor height is determined computationally with considerations to the resulting negative moment, water splash, and drag properties of our vehicle.

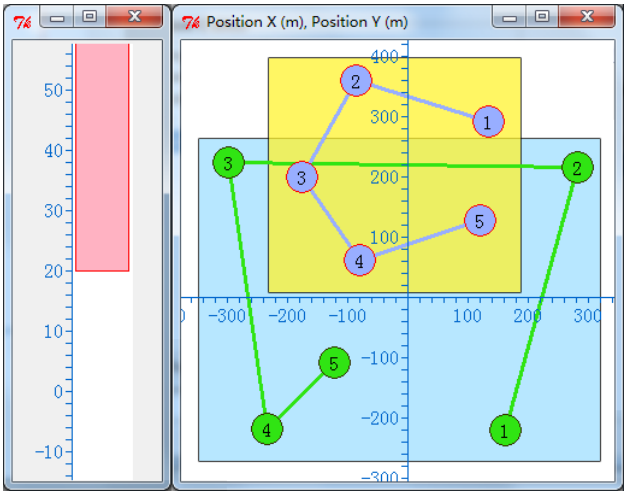
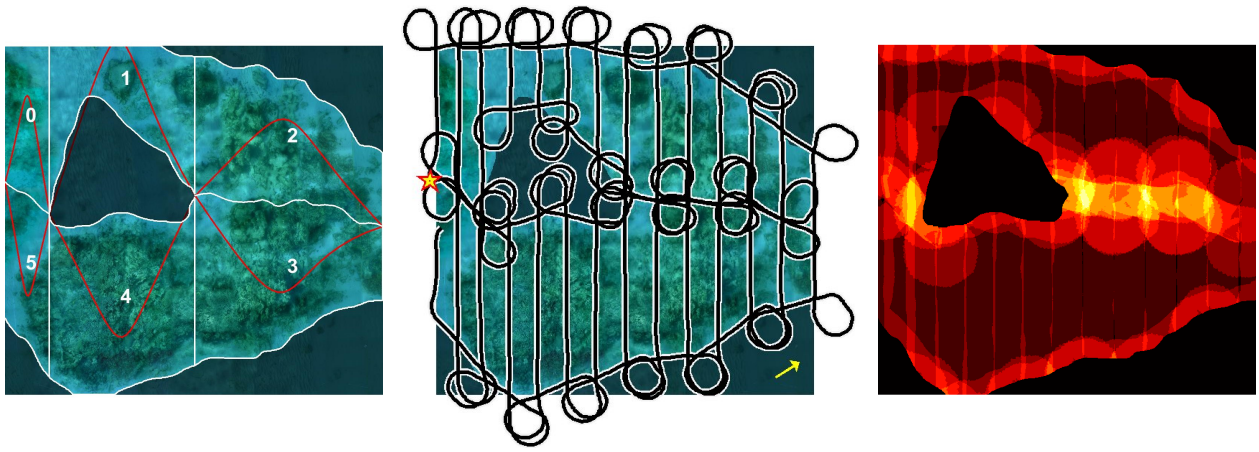


Fig. 2. Our GSSP interface [3] allows the user to visually define regional constraints (i.e. *regions*) and waypoints over different state variables. Blocks of code, written in a standard computer language, can be attached to these constructs, and are executed when the robot enters the respective region or is sufficiently close to the corresponding waypoint.

The motor controller is connected to a low-power net-book computer, which is responsible for carrying out sensor data processing, path planning, and high-level reasoning. MARE is equipped with a GPS, an IMU and a downward-facing camera as its primary sensors. Communication is achieved wirelessly through multiple media channels: WiFi is used to transfer high-bandwidth data such as a video stream in short range operations; XBee is used for mid-range (e.g. $1\,km$) and low-bandwidth external control and signaling purposes; and an analog transceiver provides emergency manual control of the boat and has the longest signal range. Sensors are powered through the net-book computer, whereas the motors are powered separately using multiple lithium polymer (LiPo) batteries. With our current configuration of two discrete $4000\,mAh$ LiPo batteries, MARE can sustain over 2 hours of continuous movement and activities.

B. Software Framework

We used a novel programming paradigm and associated software tool, called Graphical State Space Programming (GSSP) [3], to facilitate the process of specifying execution plans for our MARE platform. An execution plan can contain different components, including a sequence of waypoints depicting a high-level trajectory through the environment, a set of activities to be carried out at each location, and various reactive behaviors which may be triggered at any time during



(a) Obstacle map, cells, Eulerian circuit (b) Robot path during simulated coverage session (c) Corresponding coverage map

Fig. 3. Our terrain coverage approach [1] is demonstrated for a $200\,m \times 200\,m$ region, where the underlying coral reef image is used for illustration purpose only and is not to scale. In (a), obstacles are shown as darkened areas; white lines indicate cell boundaries resulting from the Boustrophedon decomposition process; red lines and cell numbers outline the Eulerian circuit. (b) depicts the path taken by the vehicle during a simulated coverage session at a depth of $\sim 15\,m$ and with minor current disturbances; the star denotes the starting location and the arrow indicates the average direction of current. (c) shows the corresponding coverage map, which illustrates 0.24 *percent* areas with missed coverage and 59.93 *percent* areas with repeated coverage. Missed coverage resulted due to strong currents, whereas repeated coverage were primarily caused by our *curlicue* corner-steering strategy.

the execution, such as failsafe measures. GSSP models all of these components uniformly as conditional code blocks that are triggered when certain conditions are satisfied. Code blocks are written using a standard computer language, akin to text-based programming using robot abstraction systems and development environments such as [11] & [12]. On the other hand, similar to graphical control interfaces such as [13] & [14], waypoints and regional constraints (i.e. *regions*) in GSSP are specified graphically over a topographical map of the environment (or more generally, of the state space). By combining textual programming with a graphical editor, GSSP preserves the ability to use programming constructs such as conditional statements, loops, and concurrent execution, while providing a natural visualization of the plan's components and execution flow within the state space.

As shown in Fig. 2, regions in GSSP consist of one-sided or two-sided constraints applied to one or multiple variables in the state space. GSSP also allows the user to combine multiple regions using Boolean operators (i.e. union, intersection, and negation), so as to define *Boolean regions*. Whereas regions can be specified on any state variable, waypoints on the other hand can only be defined on *writable* state variables, which are parameters that the robot's actuators can affect.

Code routines, written in Python in the current GSSP version, can be attached to any waypoint or region. An instance of the code block will be executed when the state either is sufficiently close to the respective waypoint, or satisfies the corresponding regional constraints. GSSP allows code instances from different satisfied waypoints and regions to execute concurrently, and it handles resource conflicts using a numerical priority scheme. This software framework can also regulate arbitrary binary executables that are triggered from within code blocks. These and other external applications can further interact with GSSP's internal components using a rich network programming interface.

Since plans in GSSP are specified in a robot-independent state space representation, they can be potentially deployed on different robot platforms with minimal or no adjustments needed. This abstraction layer also allows core AI routines, such as the waypoint-based motion controller, to be implemented internally in a modular manner.

IV. Coverage-Driven Exploration

Many applications require observations to be made uniformly over a region of interest. To achieve this the robot must perform coverage, by collecting sensor data continuously while following a path that sweeps through all non-obstacle locations within the desired region. This type of systematic exploration can be used to produce mosaic maps of coral reefs for example, which allows marine biologists to study these endangered ecosystems and investigate methods for their preservation.

We have adapted an autonomous terrain coverage system [1] for non-holonomic vehicles to our MARE platform. This implementation is based on an optimal and complete coverage algorithm [15] that produces a non-overlapping trajectory which completely covers a known and bounded environment while avoiding arbitrarily-shaped obstacle regions, and also terminating at the starting position. Within the marine domain, obstacles correspond to areas that we do not wish to cover, such as sand patches for instance.

A. Theoretical Formulation

Given a binary obstacle map of the coverage regions, the Boustrophedon Cellular Decomposition (BCD) technique [16] divides the free space into cells, which are interconnected by critical points at obstacle corners. Cells and critical points are stored respectively as edges and vertices within a graph structure, which is processed through a linear programming solution to the Chinese Postman Problem [17] to generate a closed cyclic path known as the Eulerian circuit,

as shown in Fig. 3(a). To ensure that the Eulerian circuit traverses through every cell exactly once, some cells are halved into non-overlapping sub-cells.

Following the ordering established by the Eulerian circuit, a complete coverage trajectory is generated by concatenating back-and-forth Seed Spreader motion paths [18] through each cell. These piecewise linear paths consist of parallel sweep lines that are uniformly separated by the footprint width. When conducting visual coverage, the footprint width should ideally correspond to the camera's field of view, so as to minimize the amount of overlap in the resulting images and thus increase the efficiency of the coverage session.

B. Pragmatic Concerns

The quality of the generated path can be improved by rotating the obstacle map prior to running the BCD algorithm, which will affect the shapes of the resulting cells. We can rotate the map to align the sweep line direction along obstacle boundaries or along the distribution of the free space. These two strategies are designed to elongate cells in order to ensure that the resulting Seed Spreader trajectory will contain long and straight paths while minimizing the number of turns. This is especially useful for non-holonomic robots such as our MARE platform, which in general exhibit poor vehicular dynamics when turning.

In the presence of strong external disturbances such as current and wind, we can alternatively rotate the obstacle map to align sweep lines with the average direction of disturbance. This ensures that the vehicle will not deviate off-course significantly when traversing through the designated path, since course deviations will prevent the camera from covering certain areas (i.e. *missed coverage*) and also result in *repeated coverage* of other areas.

Because non-holonomic vehicles cannot perform in-place rotations, we must employ additional turning maneuvers to ensure that the robot traverses through the designated piecewise linear path. Our *curlicue* strategy steers the vehicle in a circular orbit away from each turn to manually align it with the upcoming line segment, as shown in Fig. 3(b). Although this prolongs the total trajectory length, it is needed to guarantee completeness of the coverage task.

V. Surprise-Driven Exploration

On many occasions, performing a complete coverage of a region is not a suitable exploration approach. For instance, consider the case when the environment is dynamic, or when a map is unavailable, or when the region is simply too large to be covered systematically in a time-limited setting. In such scenarios, an active path planning technique, where the robot's heading is decided based on current and past observations, is arguably more useful and efficient. Hence, we propose an exploration strategy based on surprise.

Given a measure of surprise for an incoming observation, we would like to find a trajectory for the robot which maximizes the amount of surprise. We use Set Theoretic Surprise to measure the novelty of an incoming observation, and then using it to propose a greedy exploration strategy.

$(Z_{t,L}, Z_{t,R}, Z_{t,C}, Z_{t,O}) \leftarrow \text{ExtractSubImage}(Z_t)$
$\xi_L \leftarrow \xi(Z_{t,L}|\mathbf{S})$
$\xi_R \leftarrow \xi(Z_{t,R}|\mathbf{S})$
$\xi_C \leftarrow \xi(Z_{t,C}|\mathbf{S})$
if $\xi_L > \max(\xi_R, \xi_C)$ **then**
    TurnLeft()
else if $\xi_R > \max(\xi_L, \xi_C)$ **then**
    TurnRight()
else
    GoStraight()
$\mathbf{S} \leftarrow \text{UpdateOnlineSummary}(\mathbf{S}, Z_{t,O})$

Algorithm 1: SurpriseDrivenExploration $(\mathbf{S}, Z_t)$. Decides the next step for the robot, given the current observation and the current summary set.

A. Online Set Theoretic Surprise

Set Theoretic Surprise(STS) [2] is a non-parametric technique for quantifying the amount of surprise of an observation. STS has been shown to be suitable for online use [2].

At any given time, the robot maintains a set of summary samples, which are representative of all the observations made so far. These summary images not only capture mean properties of the environment, but surprising elements as well. For a new observation Z_t, we define its surprise score given the current summary $\mathbf{S} = \{S_i\}$ as:

$$\xi(Z_t|\mathbf{S}) = \min_i d(Z_t, S_i). \tag{1}$$

We update the summary if the surprise score of the observation is greater than the threshold score γ, defined as the mean score of the samples currently in the summary:

$$\gamma = \frac{1}{|\mathbf{S}|} \sum_i \min_{j, j \neq i} d(S_i, S_j). \tag{2}$$

We could either allow the summary size to grow, resulting in a summary which scales with the complexity of the data, or we could trim the summary to keep it of a constant size, by removing the summary sample with lowest surprise score given all the other summary samples.

Broder *et al.* [19] named the strategy of picking new samples above the mean or median score computed from previous picks as "Lake Wobegon" hiring strategies [1]. This strategy has been used by companies like Google and General Electric (GE) to hire a continuous stream of employees.

B. Exploration Strategy

Given a new observation and the current summary, Algorithm 1 defines a simple strategy to choose the new heading direction of the robot. At each time step, we obtain an image from the current location, and then extract four sub-images from it: top-left($Z_{t,L}$), top-right($Z_{t,R}$), top-center($Z_{t,C}$), and center($Z_{t,O}$). We compute the surprise score of each of the three top sub-images using Eq. 1, and then decide the robot's

[1]Named after the fictional town "Lake Wobegon", where according to Wikipedia "all the women are strong, the men are good looking, and all the children are above average." [19]

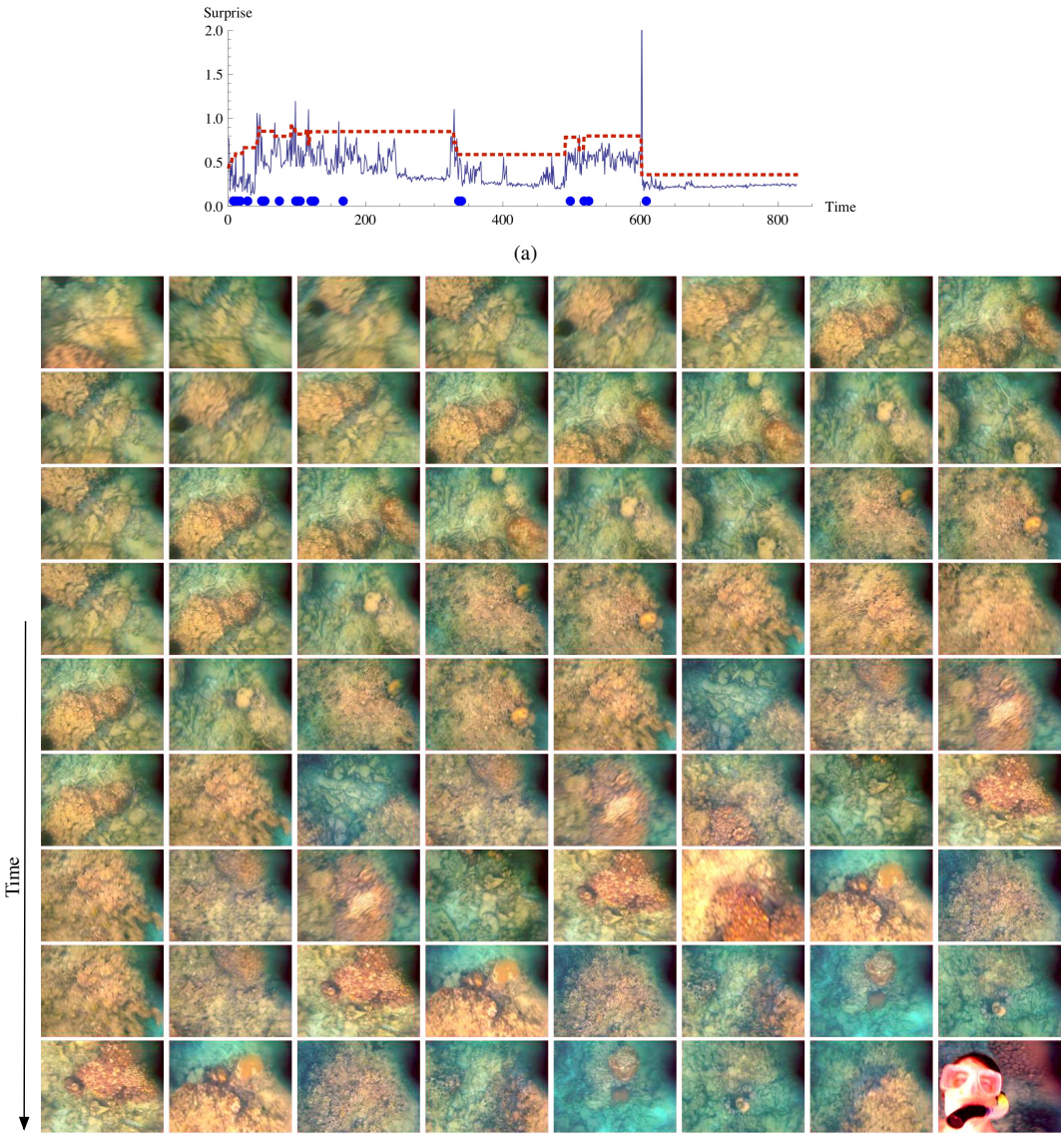


Fig. 4. Summary generated by our MARE platform as it explored a coral reef. (a) shows the surprise score (depicted as the blue line) of a new observation given the images in the summary set at that time. The dotted red line indicates the picking threshold γ. Each time the surprise score exceeds the picking threshold, the system chooses the current image and replace it with one of the images in the summary set. The picking event is depicted using blue dots. (b) shows the evolution of the summary set over time. Each row corresponds to an instance of the summary set, and hence the final row consists of the final summary set. The initial summary set shows very low diversity, although diversity increases as MARE explores the environment further.

heading based on these three scores. If the surprise score for $Z_{t,L}$ is greater than the other two, then the robot turns left; if the surprise score for $Z_{t,R}$ is greater than the other two, then the robot turns right; otherwise it moves forward along the current heading. The observation summary is updated only using the center image $Z_{t,O}$.

C. Image Representation

We use the *Bag of Words* representation of an image [20], where each image is represented by a histogram of frequency counts of visual words appearing in it.

Instead of using an offline vocabulary, we employ a dynamic online vocabulary that allows for new words to be automatically incorporated in the current vocabulary[21].

We use the Kullback-Leibler (KL) divergence metric to compute the distance between two word histograms $h_1 \& h_2$. KL divergence is not symmetric and hence is not a true distance metric, although it can be made into a distance metric by defining the distance as:

$$d(h_1, h_2) = d_{KL}(h_1||h_2) + d_{KL}(h_2||h1), \tag{3}$$

where the function $d_{KL}(\cdot||\cdot)$ computes the KL divergence between the two distributions.

VI. Results

A. Surprise-Driven Exploration

Preliminary results for our online surprise-driven exploration strategy are shown in Fig. 4. We used a constant summary size of 8 images in order to facilitate the visualization

of the results. Fig. 4(b) illustrates the evolution of the images in the summary set over time, where each row corresponds to an instance of the summary set and the last row consists of the final summary. We observe that initially the summary set has very low diversity; however, as time progresses, the samples in the summary set increase in their diversity. The final set of summary images depicts different coral species, and in particular the last image shows a diver who swam under the boat, which was correctly captured as an extremely surprising event by our summarizer.

Fig. 4(a) shows the progression of the surprise score, denoted by the blue line, of a new observation given images in the summary set at each time instance. The dotted red line illustrates the picking threshold γ. Whenever the surprise score exceeds the picking threshold, one of the images in the summary set is replaced with the current image. The picking event is indicated by the blue dots.

B. Coverage-Driven Exploration

Fig. 3(b) depicts a simulated coverage session over a coral reef region. As result of using our curlicue corner-steering strategy, the vehicle was forced to move over obstacle regions as well as previously covered cells. In the latter case, one potential heuristic to improve performance can be to switch to a greedy waypoint-based turning method whenever the curlicue orbits will result in redundant and repeated coverage. Nevertheless, for the presented obstacle configuration, our curlicue motion controller resulted in only 0.24 *percent* missed coverage, whereas steering the vehicle purely using a greedy waypoint controller following the ideal coverage path produced 0.72 *percent* missed coverage.

The coverage map in Fig. 3(c) indicates thin gaps of missed coverage between consecutive sweep lines. These areas resulted because the simulated current force constantly pushed the vehicle off-course while it was following the sweep lines. One potential method to address this issue is to dynamically reduce the footprint width during coverage based on the observed external conditions. The optimal trade-off point between the amount of missed coverage and repeated coverage will depend on the requirements of each specific application domain.

VII. Conclusion and Future Work

We have presented the *Marine Autonomous Robotic Explorer* (MARE), a novel autonomous robotic airboat with desirable mechanical properties that makes it a suitable platform for exploration-based tasks in both closed and open water environments. MARE is highly configurable and can also carry application-specific sensor payloads, making it an efficient platform choice in many applications, including autonomous surveillance, coral reef exploration, and coordination between homogeneous as well as heterogeneous robotic systems.

We have presented two different exploration strategies using MARE. The first strategy aimed at systematically covering a given region in its entirety, whereas the second strategy was driven by the goal to find surprising observations in a potentially large or highly dynamic environment. Both of these techniques have uses in a variety of applications such as mapping, surveillance, and search operations.

References

[1] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Optimal complete terrain coverage using an unmanned aerial vehicle," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA '11)*, 2011, pp. 2513–2519.

[2] Y. Girdhar and G. Dudek, "ONSUM: A system for generating online navigation summaries," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS '10)*, 2010, pp. 746–751.

[3] J. Li, A. Xu, and G. Dudek, "Graphical State Space Programming: A visual programming paradigm for robot task specification," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA '11)*, 2011, pp. 4846–4853.

[4] J. E. Manley, "Unmanned surface vehicles, 15 years of development," in *Proc. of the 2008 OCEANS Conference*, 2008, pp. 1–4.

[5] ——, "Mobile unmanned systems on, over, and under the sea," *Earth Observations*, vol. 5, no. 3, pp. 17–21.

[6] ——, "Multiple AUV missions in the national oceanic and atmospheric administration," in *Proc. of the 2004 IEEE/OES Workshop on Autonomous Underwater Vehicles*, 2004, pp. 20–25.

[7] A. Bahr and J. J. Leonard, "Cooperative localization for autonomous underwater vehicles," in *Proc. of the 10th Int. Symposium on Experimental Robotics (ISER '06)*, 2006, pp. 387–395.

[8] M. R. Benjamin, J. A. Curcio, and P. M. Newman, "Navigation of unmanned marine vehicles in accordance with the rules of the road," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA '06)*, 2006, pp. 3581–3587.

[9] B. Zhang and G. Sukhatme, "Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA '07)*. IEEE, 2007, pp. 3673–3680.

[10] A. Dhariwal and G. Sukhatme, "Experiments in robotic boat localization," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS '07)*. IEEE, 2007, pp. 1702–1708.

[11] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player/Stage project: tools for multi-robot and distributed sensor systems," *11th International Conference on Advanced Robotics*, pp. 317–323, 2003.

[12] K. Johns and T. Taylor, *Professional Microsoft Robotics Developer Studio*, ser. Wrox Programmer to Programmer. Wrox, 2008.

[13] R. Beard, D. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. McLain, and M. Goodrich, "Autonomous vehicle technologies for small fixed wing UAVs," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 2, no. 1, pp. 92–108, 2005.

[14] T. A. Kramer, R. T. Laird, M. Dinh, C. M. Barngrover, J. R. Cruickshanks, and G. A. Gilbreath, "FIRRE joint battlespace command and control system for manned and unmanned assets (JBC2S)," in *SPIE Unmanned Systems Techology VIII*, 2006.

[15] R. Mannadiar and I. Rekleitis, "Optimal coverage of a known arbitrary environment," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA '10)*, 2010, pp. 5525–5530.

[16] H. Choset and P. Pignon, "Coverage Path Planning: The Boustrophedon Cellular Decomposition," in *Proc. of the Int. Conf. on Field and Service Robotics*, 1997.

[17] J. Edmonds and E. L. Johnson, "Matching, Euler tours and the Chinese postman," *Mathematical Programming*, vol. 5, pp. 88–124, 1973.

[18] V. J. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 4, pp. 462–472, 1990.

[19] A. Z. Broder, A. Kirsch, R. Kumar, M. Mitzenmacher, E. Upfal, and S. Vassilvitskii, "The hiring problem and lake wobegon strategies," in *Proc. of the 19th ACM-SIAM Symposium on Discrete Algorithms (SODA '08)*. Society for Industrial and Applied Mathematics, 2008, pp. 1184–1193.

[20] J. Sivic and A. Zisserman, "Video google: Efficient visual search of videos," in *Toward Category-Level Object Recognition*, ser. Lecture Notes in Computer Science, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Eds. Springer Berlin / Heidelberg, 2006, vol. 4170, pp. 127–144. [Online]. Available: http://dx.doi.org/10.1007/11957959_7

[21] Y. Girdhar and G. Dudek, "Online visual vocabularies," in *Proc. of the 8th Canadian Conference on Computer and Robot Vision (CRV '11)*. IEEE Computer Society, 2011, pp. 191–196.