PROPOSED ALGORITHM FOR AUTONOMOUS NAVIGATION IN THE LITTORAL ZONE FOR AMPHIBIOUS ROBOTS

Mark Borg[1], Scott B. Nokleby[1]
[1]*Faculty of Engineering and Applied Science, University of Ontario Institute of Technology, Oshawa, Canada*
Email: mark.borg@uoit.ca; scott.nokleby@uoit.ca

ABSTRACT

This paper proposes an algorithm for goal planning and obstacle avoidance for autonomous, amphibious robots within the littoral zone. The littoral zone is the transition area between land and water that poses many challenges to autonomous robot navigation. The algorithm is presented in detail. Testing of the algorithm using a land-based autonomous robot set-up was done as well as testing of the algorithm on actual littoral zone images. This work represents an important step toward developing a robot that can navigate autonomously through the littoral zone.

Keywords: amphibious; autonomous; littoral.

ALGORITHME POUR LA NAVIGATION AUTONOME DANS LA ZONE LITTORALE POUR DES ROBOTS AMPHIBIES

RÉSUMÉ

Cet article propose un algorithme pour la planification d'objectifs et l'évitement d'obstacles pour les robots autonomes et amphibies dans la zone littorale. La zone littorale est la zone de transition entre la terre et l'eau qui pose de nombreux défis à la navigation autonome des robots. L'algorithme est présenté en détail. Des tests de l'algorithme utilisant une configuration de robot autonome utilisée sur terre ont été effectués, ainsi que des tests de l'algorithme sur des images réelles de la zone littorale. Ce travail représente une étape importante dans le développement d'un robot capable de naviguer de manière autonome dans la zone littorale.

Mots-clés : amphibie ; autonome ; littorale.

1. INTRODUCTION

Navigation of a robot in the tidal area or littoral zone is notoriously difficult as it is an obstacle rich environment that is constantly in flux with multiple variables. The littoral zone is described as the area close to and including the shoreline as shown in Figure 1. The littoral zone variables include the location of the water's edge which fluctuates due to waves, wind, surge, and tidal flow. Other complications in identifying obstacles in this area include light refraction, reflection, flotsam, moving obstacles such as humans/animals, and beach debris. This expansive spectrum of environments and obstacles must be factored into account when an amphibious, autonomous robot is attempting to navigate through the area.

Research into autonomous navigation, in an obstacle rich environment, with imperfect sensor data is limited. Xu, et al. [1] developed an algorithm for planetary landers in unknown environments. They performed simulation results of autonomous navigation based on sequential images taken with an Inertial Measurement Unit (IMU) and a monocular camera. This work is significant due to its ability to make decisions regarding the landing area based from a limited amount of information from LiDAR (Light Detection and Ranging) sensor data and a monocular overhead image. Although a promising concept, the work remains at the simulation stage. Underwater robotic research performed by She and Tian [2] looked at using an algorithm that utilizes sensors and cameras to monitor static and dynamic underwater obstacles. This research focused on an algorithm to obtain position, speed, and acceleration information from the position of obstacles located by the camera and sensors in real-time. This is of interest to amphibious work since underwater robots must also deal with drift which is a concern for floating robots. This has also only been tested in simulation. The amphibious snake robot researched by Zhao, et al. [3] uses kinematics alone to self localize. Prototype work has shown that the snake's mechanical arrangement and an IMU were satisfactory to allow the snake to localize. The benefit of this type of system is that it is extremely efficient in power usage. This work has only been tested with the robot on land and not water where drift will complicate the positioning algorithm. Ragi, et al. [4] developed an algorithm that takes data from multiple sensors on an amphibious robot and fuses it together so that the algorithm can track multiple points, representing multiple victims along a flowing river, during an emergency. The benefit of this algorithm is its ability of tracking multiple sources of data from multiple sensor types to give a single outcome. This algorithm remains in the simulation stage of research.

The spherical robot designed by Pan, et al. [5] uses a colour image derived from a Microsoft Kinect™ sensor. The algorithm developed utilizes a Kalman estimation mechanism to process colour images from RGB-D camera images. The benefit of this system is that it is easy to implement. The drawback of this type of setup is that it is not waterproof or robust. A simple and robust sensor system that was proposed

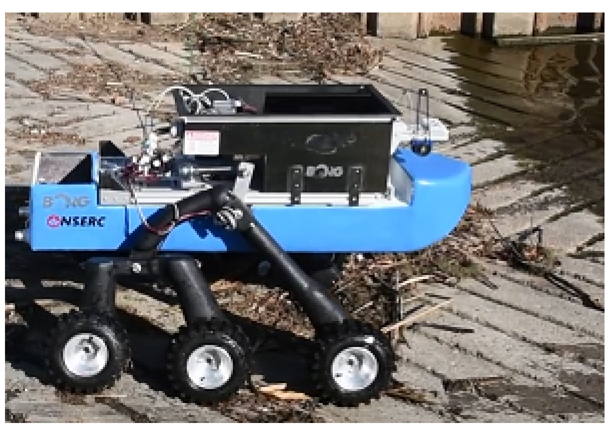

Fig. 1. Autonomous Amphibious Robot (AAR) robot navigating through the littoral zone.

for an autonomous, amphibious vehicle was proposed by Frejek and Nokleby [6]. This design is based on a four-wheeled, articulated platform that utilized five ultrasonic sensors, a GPS, and a microcontroller. The benefit of this type of simple, stand alone system is that it does not require outside communication of data. The drawback to this sensor array is that it cannot path plan large distances because of the proximity of the sensors to the ground. Consi, et al. [7] developed a navigation system that monitors positioning by GPS and an IMU. These two sensors track slippage and deformation of the beach substrate such as sand. Testing indicated that the robot was not large enough to keep from becoming stalled in the sand. A cockroach inspired, six-legged, robot for surf zone operations designed by Harkins, et al. [8] offers a novel way of maneuvering over obstacles in the surf zone. The articulated body of the robot and six-rotating legs, called Wheggs™, allow the robot to climb over many obstacles. This climbing ability reduces the need to navigate around some obstacles.

There are numerous other papers in the literature on amphibious robots (see for example [9–11]). However, the majority of papers in the literature focus on the design and locomotion of amphibious robots, not on autonomous navigation.

Due to the complexity of the environment and the types of obstacles that can be encountered, amphibious, autonomous robots have a considerable amount of sensor data to identify. Dependant on the size of the amphibious robot, the sensor data tends to be close to the water which must contend with waves, debris, and light reflection off of the water. This is complicated when the robot is on land and the sensors are now at a different height to the environment at which the robot is operating on. The robot also needs to recognize when it is in water, since the water can confuse the sensors. The parameters of how the robot progresses and performs when navigating over/through water is significantly different than when operating on land.

A robot platform, dubbed the Autonomous Amphibious Robot (AAR), is being developed for tasks such as search and rescue, natural disaster support, research, environmental monitoring, and mapping at the Mechatronic and Robotic Systems Laboratory at the University of Ontario Institute of Technology [12]. The abilities of the AAR were considered when creating the proposed algorithm for autonomous navigation. Acquiring situation data of the littoral zone can be difficult if the sensors are only located on the robot due to the limited field of view that the sensors have, especially when the robot is in the water. Increasing the field of view to above the robot is needed so that the entire littoral zone can be considered by the navigation algorithm. This allows for one overall path planning operation to take place without multiple fine adjustment course corrections which can be difficult while navigating over water. Aerial imaging of the littoral zone captured by an Unmanned Aerial Vehicle (UAV) is used by the algorithm to find an obstacle free path through the littoral zone.

This paper presents the proposed algorithm for autonomous navigation of amphibious robots through the littoral zone using overhead imagining. Section 2 presents the algorithm. Section 3 shows preliminary land-based testing of the algorithm using a Pioneer P3-DX. Section 4 presents test results for actual littoral zone images. Section 5 presents future work and Section 6 presents the conclusions for the paper.

2. PROPOSED LITTORAL ZONE NAVIGATION ALGORITHM

The littoral area is a data rich environment. Early testing for the project found that sensors on the robot gave only 1 m of visibility on water and 5 m on land. This was due to the AAR sensor position being at 0.9 m when on land and 0.4 m when the AAR is in the water. At that point, it was decided that an image from above the littoral zone would give the full field of view of the area which would allow the AAR to build a cost map of the littoral zone so that an obstacle free path could be navigated.

When approaching the littoral zone the AAR will stop at a preset GPS waypoint and launch a UAV. The UAV will hover above the GPS waypoint and capture images of the littoral zone including the pose of the QR code on the AAR relative to the littoral zone and any perceived obstacles. The waypoint when approaching

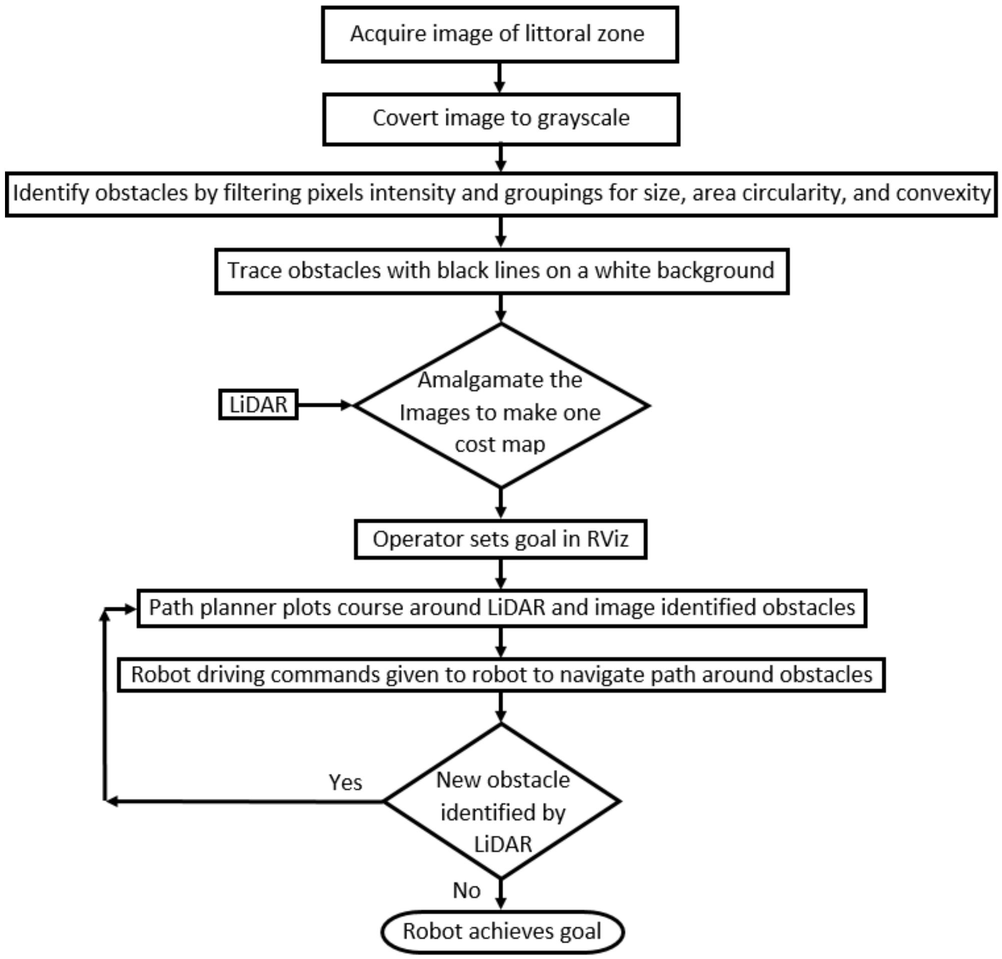

Fig. 2. Algorithm flow chart.

the littoral zone was set at 10 m from the shoreline. This gave a position for the robot that would not include water when approaching from land and not include land when approaching from the water. At this location the robot will launch a UAV, have it hover ∼15 m above and ∼10 m ahead of the AAR and have it take images of the littoral zone. These images would be transferred back to the AAR for adjustments to remove any distortion that may be introduced by the fish eye lens. The navigation algorithm then can make an assessment of the entire littoral zone around the robot and plot a course to avoid obstacles. The AAR will have a 0.7 m x 0.7 m QR code mounted on the robot that will act as a launching/landing pad for the UAV. The QR code parameters are loaded into the algorithm so that the algorithm can use the relative size and angle of this image as a reference position. This reference allows the algorithm to calculate the position of the UAV relative to the camera and other objects in the image. The algorithm then builds a cost map of the littoral zone, the position of the AAR, and the relative position of the perceived obstacles. A flow chart of the algorithm is shown in Figure 2. Initial testing of the algorithm used a Pioneer P3-DX mobile robot instead of the AAR to facilitate the development of the algorithm.

The coding language Python was utilized in the development of the algorithm. The Python code is run on the Robotic Operating Software (ROS - www.ros.org) which runs on the LINUX operating system. Subroutines for the controls of the Pioneer P3-DX robot are accomplished with the `p2os` driver package

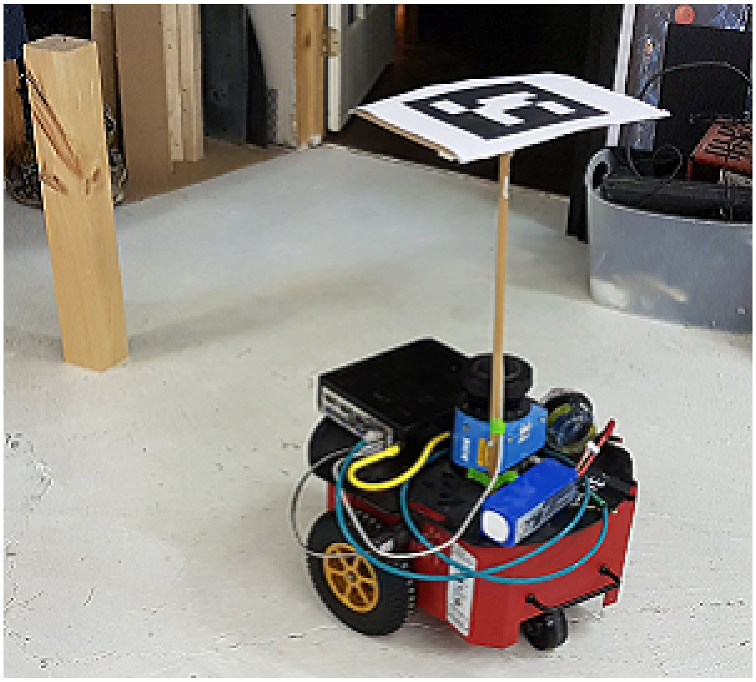

Fig. 3. Pioneer P3-DX with QR Code.

within ROS. The LiDAR package `LMS1xx` was used. For the computer vision library, OpenCV was utilized in ROS. OpenCV provides computer vision algorithms and utilities. For QR code identification the `ar` single board package was utilized. Laser-based Simultaneous Localization and Mapping (SLAM) was provided by the ROS node `gmapping`. For visualization of the robot and 2D navigation goal setting the ROS package `RViz` was used. The Canny Edge detection algorithm was used to trace obstacles.

There are six modules to the algorithm: QR identification; localization of the QR code in relation to the image source and the obstacles identified in the image; the identification of land, underwater, and floating obstacles; robot path navigation; construction of the cost map from the LiDAR data; and immediate course adjustment for collision with impending obstacles such as humans. The algorithm follows the flow chart as shown in Figure 2 and specific details are discussed in the next section.

3. TEST RESULTS WITH THE PIONEER P3-DX

Due to the physical size of the AAR and the limitations that testing outdoors involves, a Pioneer P3-DX mobile robot was used for preliminary development of the algorithm and to test the algorithm indoors. A QR code was mounted on top of the Pioneer P3-DX (see Figure 3) and an obstacle course was devised. Overhead image data (obtained from a fixed camera) as well as LiDAR data were combined into one cost map. A 2D navigational path through the obstacle field was tested using this experimental setup.

3.1. QR Code Identification

The identification of the QR code was crucial to the design of the algorithm since the position of the robot relative to the position of the obstacles was required so that an occupancy grid could be built and an obstacle free path determined. The Open Source Computer Vision Library (OpenCV - opencv.org) for Python was utilized as the base code for this portion of the algorithm. To capture an appropriate sized area of the littoral zone, testing revealed that an image acquired by a camera from $\sim$15 m above the water's edge would capture the full area that needed to be navigated. The image is collected by a quad-copter UAV.

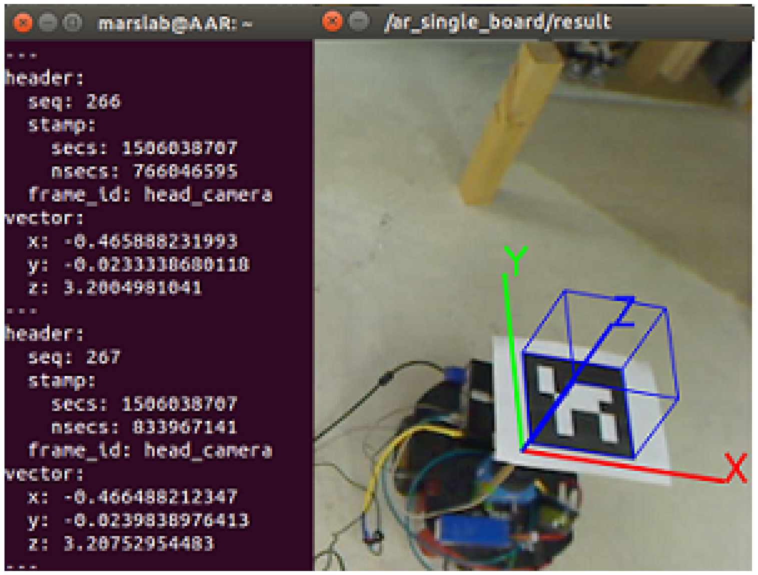

Fig. 4. Identification of the QR Code Mounted on the Pioneer P3-DX Robot.

Multiple sized QR codes were tested before it was found that a 0.7 m x 0.7 m QR code was optimum to be consistently identified from that height. For the indoor portion of the testing, the image was taken from 2.1 m above and the QR code measured 0.17 m x 0.17 m as shown in Figure 4. The OpenCV code for QR identification gives the position of the QR code in the image. This position information is logged in a separate file. This position data is used in the alignment of the overhead obstacle position image with the LiDAR streamed data.

3.2. Construction of the Cost Map from the LiDAR Data

The initial cost map created in ROS is built from the LiDAR data and visualized in `RViz` through ROS as shown in Figure 5. The ROS stack allows the LiDAR data to be recognized as occupied cells and creates an occupancy grid. Images that are dense and do not vary after 3 seconds, such as walls, are recognized by the SLAM node as structures. Moving objects, such as humans, are recognized as such and are identified.

3.3. Identification of Obstacles from Overhead Images

The blob detection code was written in the Python language utilizing the OpenCV package. OpenCV identifies differences in pixel intensity to classify possible obstacles. To accomplish this, the first step in the algorithm is to take the overhead image as seen in Figure 6 which will be captured by the UAV hovering above the robot. Once the image is obtained it is converted from colour into a gray-scale image. This allows contrast to be used as a first step in obstacle identification. The second step is to compare all of the pixels while using a set of filters that categorize for size, area circularity, and convexity. Once the obstacles are identified, the third step identifies an appropriately sized rectangular area to encompass the obstacle. This obstacle area is completed for each obstacle found on the image. The image background is removed only leaving the obstacles in the image as seen in Figure 7. For fourth step, the Canny Edge detection ROS node is utilized to edge detect the obstacles and line trace them. The fifth step flips the black background to a white background and converts the line traced obstacles to black as seen in Figure 8. With this data combined with the LiDAR sensor data, the `p2os` navigation package within ROS plots a course around the obstacles to the path goal as shown in Figure 9.

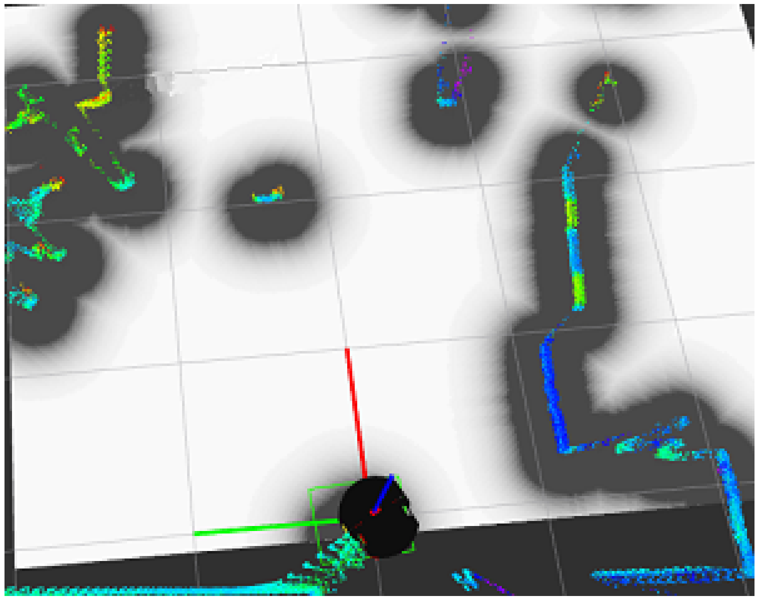

Fig. 5. Example of a cost map created by the LiDAR sensor mounted on the Pioneer P3-DX.

3.4. Robot Path Navigation

The testing of the robot path navigation was performed with the Pioneer P3-DX mobile robot. The collection of drivers was derived from the ROS node gmapping. This allows for control of the robot drivers. Visualization and 2D navigational goal data was given through the RViz visualization package in ROS. An obstacle course was setup to test the algorithm as shown in Figure 6.

3.5. Collision Avoidance

Collision avoidance was accomplished by combining the data brought in by the overhead image, the location of the QR code, and the LiDAR developed cost map. The amalgamation of this data was integrated into one cost map. The occupied cells were given an inflation factor so that the robot would maintain a safe distance from them. A navigational goal is set by the operator and is shown in Figure 9 as a green arrow. The navigation algorithm determined a safe path around the obstacles and is indicated by the purple line in Figure 9. The active LiDAR data stream also gives real-time collision avoidance to dynamic obstacles such as a human walking in front of the moving robot.

Results of the algorithm defining the path of the Pioneer P3-DX robot within the cost map created by a LiDAR sensor, as well as the amalgamation of the overhead obstacle identified image, showed that the algorithm was able to consistently identify a path that was free from obstacles, even when the obstacles were below the plane of the 2D LiDAR sensor. The robot also responded to real-time collision avoidance and recalculated the path to the waypoint that was obstacle free. The algorithm inflated safety zones around both the obstacles identified by the LiDAR and the overhead image.

4. RESULTS WITH ACTUAL LITTORAL ZONE IMAGES

To test the algorithm in real-world conditions, images were acquired from a number of littoral areas on multiple water bodies. Sun angle, cloud cover, weather, and season were also varied. This was done to vary the images so that the robustness of the algorithm could be established. A variety of beach base materials were tested. This included rocks, sand, pebbles, and grass. The lake bottoms and flotsam were also varied to test the robustness of the algorithm in identifying obstacles. These obstacles included logs, plastic waste,

Fig. 6. Overhead image of the test area with two obstacles below the LiDAR sensor range and one obstacle within range.

tires, and seaweed. Initial results identified that the algorithm had a 27% false positive rate. It was found that the filter setting that identified obstacles was too fine so it was adjusted for larger objects. This change allowed the false positive rate to be reduced to 14%. Further testing indicated that light coloured objects that should have been classified as obstacles were not being identified. A correction was made to the filter that controls contrast which reduced the false positive rate to less than 10%.

Figures 10 and 11 show two examples of the effectiveness of the algorithm used to process the overhead images for real-world littoral images. The littoral zones feature both natural and man made obstacles. For both figures, the left-hand side shows the initial littoral image taken with a UAV and the right-hand side shows the processed image with the identified obstacles traced. The figures show that the algorithm is able to identify a variety of obstacles.

5. FUTURE WORK WITH THE AAR

The AAR is a six-wheeled amphibious robot with dual water jet drives for water propulsion (see Figures 1 and 12 [12]). It has been designed and developed to be able to handle obstacles that are typical of the littoral zone. The navigation algorithm is currently configured for the testing that was conducted utilizing the two-wheel drive Pioneer P3-DX. The algorithm will be adjusted to suit the six-wheel drive larger platform of the AAR. The algorithm will be amended for the larger footprint of the AAR and the climbing characteristics of the amphibious robot. The software will accommodate input from sensors to determine when the AAR has entered the water. The algorithm will have the steering controls augmented to suit the AAR's aquatic mode jet drives.

The task for the finalized AAR will have it start at a predefined waypoint on the beach strewn with obstructions. The obstructions that are larger than 0.5 m x 0.5 m x 0.5 m will be considered by the algorithm as obstacles and identified as needing to be avoided by the AAR. A quad-copter UAV will fly above the littoral zone and an image will be taken. This image will be uploaded into the algorithm and will have

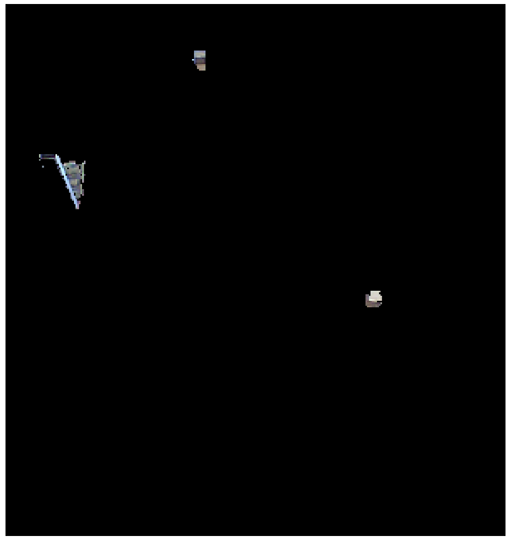

Fig. 7. Image with obstacles identified with background removed.

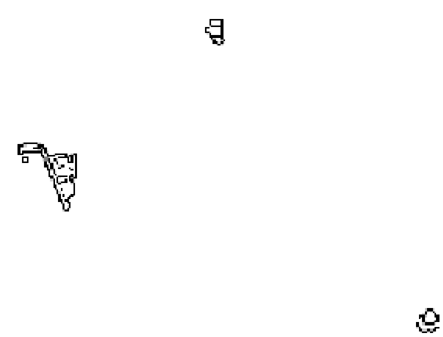

Fig. 8. Image stripped of all data other then the edge detection of the obstacles.

the obstructions filtered and the obstacles identified. A cost map will be created with the obstacles marked as occupied cells. An additional cost map will be created with real-time LiDAR data. The algorithm will amalgamate the two cost maps and will create a path to a goal 10 m past the water's edge which has been predefined by a user. This point in the water is a staging point for the AAR's next navigational waypoint. The algorithm will continuously update the path to the goal with LiDAR data to accommodate for new obstacle contacts. As the AAR transitions to water, sensors on the AAR will engage the water jet drives as well as the wheels to allow it to transition to full water propulsion. Once at the target waypoint, the AAR wheels will stop rotating and it will be ready for the next waypoint in water propulsion mode only.

6. CONCLUSIONS

An algorithm is proposed to determine an obstacle free path through the littoral zone for autonomous amphibious robots. Tests with a Pioneer P3-DX show that the proposed algorithm is capable of identifying a QR code and determining the position of the QR code in the image by the angle and size of the QR code. The algorithm is also able to identify obstacles based on developed filters and output a file that lists the positions of all of the obstacles based on their relative position to the QR code via an occupancy

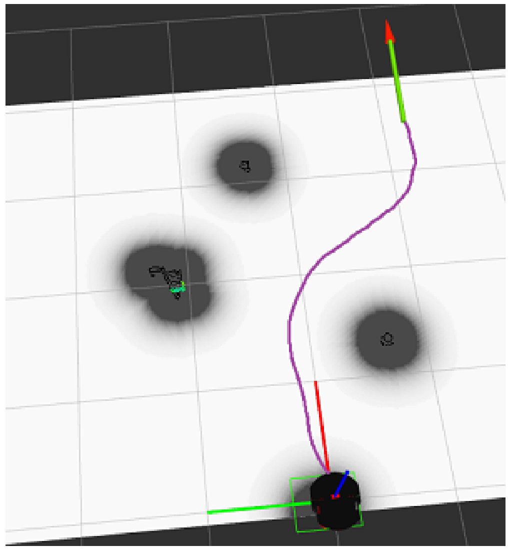

Fig. 9. 2D navigational goal (green arrow) and calculated path (purple line) based on LiDAR and overhead image data.

Fig. 10. Original littoral image (left) with processed image output (right).

grid. The algorithm can amalgamate the cost map built by the LiDAR data with the occupancy grid made from the overhead image and determine the best path through the obstacles to the navigational goal set by the operator. Testing of the algorithm with real-world littoral zone images shows the effectiveness of the proposed approach.

ACKNOWLEDGEMENTS

Professor Roger Boudreau, Université de Moncton, is thanked for doing the French translation of the paper title and abstract.

REFERENCES

1. Xu, C., Wang, D. and Huang, X. "Autonomous Navigation Based on Sequential Images for Planetary Landing in Unknown Environments." *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 10, pp. 2587–2602, 2017.
2. She, M. and Tian, L. "A Novel Path Control Algorithm for Networked Underwater Robot." *Journal of Robotics*, pp. 1–7, 2018.
3. Zhao, X., Dou, L., Su, Z. and Liu, N. "Study of the Navigation Method for a Snake Robot Based on the Kinematics Model with MEMS IMU." *Sensors*, Vol. 18, No. 3, pp. 879–902, 2018.

Fig. 11. Original littoral image with numerous man-made and natural obstacles (left) with processed image output (right).

Fig. 12. The AAR in Lake Ontario.

4. Ragi, S., Tan, C.S. and Chong, E. "Guidance of Autonomous Amphibious Vehicles for Flood Rescue Support." *Mathematical Problems in Engineering*, pp. 1–9, 2013.
5. Pan, S. Shi, L. and Guo, S. "A Kinect-Based Real Time Compressive Tracking Prototype System for Amphibious Spherical Robots." *Sensors*, Vol. 15, No. 4, pp. 8232–8252, 2015.
6. Frejek, M. and Nokleby, S.B. "Design of a Small-Scale Autonomous Amphibious Vehicle." In "Proceedings of the 21st IEEE Canadian Conference on Electrical and Computer Engineering," pp. 781–786. Niagara Falls, Canada, May 2008.
7. Consi, T.R., Ardaugh, B.R., Erdmann, T., Matsen, M., Peterson, M., Ringstad, J., Vechart, A. and Verink, C. "An Amphibious Robot for Surf Zone Science and Environmental Monitoring." In "Proceedings of OCEANS 2009," pp. 1–7. Biloxi, USA, October 2009.
8. Harkins, R., Ward, J., Vaidyanathan, R., Boxerbaum, A.S. and Quinn, R. "Design of an Autonomous Amphibious Robot for Surf Zone Operations: Part II - Hardware, Control Implementation and Simulation." In "Proceedings of the IEEE International Conference on Advanced Intelligent Mechatronics," pp. 1465–1470. Monterey, USA, July 2005.
9. Yu, J., Ding, R., Yang, Q., Tan, M., Wang, W. and Zhang, J. "On a Bio-inspired Amphibious Robot Capable of Multimodal Motion." *IEEE/ASME Transactions on Mechatronics*, Vol. 17, No. 5, pp. 847–856, Oct 2012.
10. Klein, M.A., Boxerbaum, A.S., Quinn, R.D., Harkins, R. and Vaidyanathan, R. "SeaDog: A Rugged Mobile Robot for Surf-Zone Applications." In "Proceedings of the 4th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)," pp. 1335–1340. Rome, Italy, June 2012.
11. Crespi, A., Karakasiliotis, K., Guignard, A. and Ijspeert, A.J. "Salamandra Robotica II: An Amphibious Robot to Study Salamander-Like Swimming and Walking Gaits." *IEEE Transactions on Robotics*, Vol. 29, No. 2, pp. 308–320, April 2013.
12. Borg, M., Nokleby, S.B., Puccini, L., Mezil, A., Lau, B., Dobrescu, O., Idris, J., Chan, A., Cain, C., Long, K., Thaker, K. and Lam, K. "Design, Development, and Preliminary Testing of an Autonomous Amphibious Robot." In "Proceedings of the 2016 CSME International Congress," pp. 1–6. Kelowna, Canada, June 2016.