Real-Time Monocular Obstacle Avoidance using Underwater Dark Channel Prior

Paulo Drews-Jr$^{1,2,3}$ Emili Hernández$^1$ Alberto Elfes$^1$ Erickson R. Nascimento$^2$ Mario Campos$^2$

Abstract—In this paper we propose a new vision-based obstacle avoidance strategy using the Underwater Dark Channel Prior (UDCP) that can be applied to any Unmanned Underwater Vehicle (UUV) equipped with a single monocular camera and minimal on-board processing capabilities. For each incoming image, our method first computes a relative depth map to estimate the obstacles nearby. Then, the map is segmented and the most promising Region of Interest (RoI) is identified. Finally, an escape direction is computed within the RoI and a control action is performed accordingly to avoid the obstacles. We tested our approach on a video sequence in a natural environment and compared it against a state-of-the-art method showing better performance, especially in light changing conditions. We also provide online results on a low-cost Remotely Operated Vehicle (ROV) in a controlled environment.

I. INTRODUCTION

In the last few years there has been an increase of Unmanned Underwater Vehicles (UUVs) available to the general public. These modern vehicles are different from the traditional commercially available ones and those built for research purposes as they tend to be small, affordable and with highly limited sensing capabilities. One example is the OpenROV$^1$, which has a color camera onboard in its standard configuration.

Vision-based sensors have been extensively used in many underwater robotic applications such as habitat and animal classification$^2$, mapping$^3$, 3D scene reconstruction$^4$, visualization$^5$, docking$^6$, tracking$^7$, inspection$^8$ and robot localization$^9$. However, very few works have addressed the vision-based obstacle avoidance problem in the underwater domain as it is usually solved with sonar-based sensors$^{10}$. The work by Roser et al.$^{11}$ is based on binocular vision. The main limitation of their method is the requirement of a calibrated stereo pair and the associated computational cost. Recently, Rodríguez-Telles et al.$^{12}$ proposed a method to avoid obstacles using a monocular camera that requires an offline learning phase and superpixel based segmentation$^{13}$. The training step is used to obtain the water color which is then generalized to the whole dataset. The main drawback of this approach is the highly dependence on the medium conditions and it often requires specific training and manual tuning of the algorithm’s parameters for each dataset.

In this paper, we propose a real-time monocular obstacle avoidance method suitable for small ROVs. Our approach estimates a depth map using statistical priors$^{14},^{15}$ and a physical underwater light attenuation model. Differently from the images captured in the air, the underwater images carry on information about depth because of the relation between depth and medium effects. Thus, we explore this property using a statistical prior to estimate the depth map. Then, it is segmented using an adaptive threshold and a set of Regions of Interest (RoIs) are identified based on an ellipse fitting technique. We also compute an escape direction using the center of mass of the most promising RoI that avoids the collision with the nearby obstacles. Finally, we use a simple but effective control strategy to turn the direction vector between the robot and the escape direction into thrusters setpoints encoded as Pulse Width Modulation signals (PWMs).

The main contribution of this work is an underwater obstacle avoidance method that achieves real-time performance using monocular images. We applied a statistical model-based depth map estimation for obstacle avoidance purposes. We also present results using offline experiments in real oceanic condition and compared it against Rodríguez-Telles et al.$^{12}$ showing better results. Furthermore, we show that our algorithm achieves real-time performance in an OpenROV, a low-cost ROV equipped with a single camera, in a controlled environment.

The remainder of the paper is organized as follows: Section II describes the proposed obstacle avoidance method; Section III evaluates the methodology using experimental field data; finally, in Section IV, we summarize the paper contributions and draw the future research directions.

II. METHODOLOGY

Our approach uses images from a single monocular color camera and it generates a depth map of the scene based on a light attenuation model and statistical priors. In fact, light is absorbed and scattered by the medium before reaching the camera, and the understanding of these effects allow us to estimate the depth map. This depth map is segmented into RoIs which allows us to compute an escape direction. This is turned into control set-points and feed directly to the control of the each thruster. Fig. 1 shows the main steps of
our method, and Fig. 2 depicts the intermediate steps for a single frame.

A. Monocular Depth Map Estimation

1) Physical Underwater Light Attenuation Model: Underwater images are the result of a complex light rays interaction, the medium and the scene structure. Jaffe-McGlamery proposed one of the most used model to describe this interaction [16], [17], in which the image intensity is composed of three terms: the direct illumination ($E_d$), the forward-scattering ($E_{fs}$) and the backscattering ($E_{bs}$):

$$E_T = E_d + E_{fs} + E_{bs}. \quad (1)$$

Part of the light radiated from objects is scattered and absorbed by the medium and the remaining portion, called direct illumination, reaches the sensor. Direct illumination [17] is formulated as:

$$E_d = J e^{-\eta d} = J t_r, \quad (2)$$

where $J$ is the scene radiance, $d$ is the depth, and $\eta$ is the attenuation coefficient. The attenuation coefficient $\eta$ is composed of the scattering and the absorption coefficients, both wavelength dependent [18]. $t_r$ is the medium transmission, modeled as the exponential term.

Since the backscattering $E_{bs}$ is the main reason for image contrast degradation in most of cases, forward scattering $E_{fs}$ is usually neglected [19]. The backscattering does not originate from the object’s radiance, but it results from the interaction between the sources of ambient illumination with particles dispersed in the medium. A simplified model for the $E_{bs}$ component can be described as:

$$E_{bs} = A(1 - e^{-\eta d}) = A(1 - t_r), \quad (3)$$

where $A$ is the global light which is wavelength dependent. This is estimated by finding the brightest pixel in the darkness channel [20]. The other terms are the same as for the direct components.

The final model describing the formation of an image $I$ acquired in an underwater homogeneous medium with natural light can be formulated as:

$$I(x) = J(x) t_r(x) + A(1 - t_r(x)), \quad (4)$$

where $x$ are the pixel coordinates.

2) Transmission Prior and Depth Estimation: The image formation model described in Eq. 4 is an ill-posed problem, since it is not possible to solve the depth ($d$) and the true appearance of the scene ($J$) without prior knowledge about the scene.

[21] proposed the Dark Channel Prior (DCP), a statistical prior based on the observation that natural images exhibited a mostly dark intensity in a square patch in at least one color channel of the image. It is difficult to validate this assumption and the corresponding statistic correlation in underwater images due to the impossibility to obtain real underwater images without the medium. Despite this difficulty, the main assumption stated by [21] is still plausible for which at least one color channel has some pixels whose intensity are close to zero. This low intensity pixels are due to shadows, objects or surfaces where at least one color channel has low intensity, like fishes, algae or corals, and dark objects or surfaces like rocks or dark sediment.

However, the wavelength independence claim is false in most of the cases due to the high absorption rates in the red channel in typical oceanic conditions. Hence, we adopted a prior called Underwater Dark Channel Prior (UDCP) [14], [15]:

$$J^{UDCP}(x) = \min_{y \in \Omega(x)} (\min_{c \in G,B} I_c^r(y)). \quad (5)$$

Considering Eq. 4 and the UDCP assumption, it is possible to isolate the transmission $t_r$ in a local patch $\Omega$. Applying the minimum operation to both sides, we can estimate $t_r$ based on the image $I$ and the global light $A$ as:

$$\hat{t}_r(x) = 1 - \min_{y \in \Omega(x)} (\min_{c \in G,B} \frac{I_c^r(y)}{A^c}), \quad (6)$$

where the global light $A$ is estimated by finding the brightest pixel in the $J^{UDCP}$ (Eq. 5). [15] provides a experimental verification of the UDCP assumption and more details about its applicability.

We define the square patch $\Omega = 15 \times 15$ for $640 \times 360$ pixels images. The minimum operator is similar to the classical erosion morphological operator. Thus, we compute the minimum filter using a fast operator as proposed in [22], with a linear complexity with respect to the image size. Fig. 2b depicts an example of a transmission map $\hat{t}_r$.

Based on the transmission map, we can estimate the depth map $D$ up to the unknown attenuation coefficient $\eta$, as:

$$\hat{D}(x) = \eta d(x) = -\log \hat{t}_r(x). \quad (7)$$

In the actual implementation, the log operator is computed based on LookUp Tables (LUT) to improve the performance. Differently from image restoration works, we do not perform any refinement procedure due to time constraints. The depth map obtained is adequate for robotics tasks such as obstacle avoidance. However, some filtering operations are performed
C. Escape Direction Estimation and Control Scheme

In blue, and the others, in green. within all neighboring pixels. Fig. 2d shows the largest RoI, methods [23]. RoIs are estimated based on segmented pixels the obstacle extent, typically performed on path planning this operation is similar to the one produced by increasing a circular kernel with radius \( r \). we apply an erode operation in the segmented pixels using the ellipse shape, a circle of radius \( r \) least squares optimization [24] in the largest RoI. Based on circle with radius \( r \) according to their size and those smaller than the area of the illumination changes [21].

If it is possible to fit in it a circle of radius \( r \), we fit an ellipse using to improve the segmentation step: a median filter using a kernel of \( 5 \times 5 \) pixels, and a Gaussian filter with the same size of \( \Omega \). Fig. 2c illustrates the final depth map.

B. Segmentation

For each incoming depth map, we first perform a binary segmentation. The threshold level is estimated as a fraction of the global light \( A \). This simple approach is robust to light variation because \( A \) changes according to the illumination of each frame. Thus, the segmentation is partially invariant to illumination changes [21].

Similar to [12], we assume that a RoI is safe for the robot if it is possible to fit in it a circle of radius \( r \). Therefore, we apply an erode operation in the segmented pixels using a circular kernel with radius \( r \). The effect generated by this operation is similar to the one produced by increasing the obstacle extent, typically performed on path planning methods [23]. RoIs are estimated based on segmented pixels within all neighboring pixels. Fig. 2d shows the largest RoI, in blue, and the others, in green.

C. Escape Direction Estimation and Control Scheme

1) Escape Direction: The RoIs obtained are sorted according to their size and those smaller than the area of the circle with radius \( r \) are removed. Then, we fit an ellipse using least squares optimization [24] in the largest RoI. Based on the ellipse shape, a circle of radius \( r \) is fitted within the ellipse at the RoI center of mass (see Fig.2e). If the circle is contained in the ellipse, it is accepted as an escape direction. Otherwise, this process is repeated for the next valid RoI until a suitable escape direction is found. The radius \( r \) is empirically estimated as its value depends on the camera, the robot and the environment.

As proposed in [12], the pitch angle is set to an upward direction based on the camera’s field of view in case a valid escape direction is not found.

In order to prevent sudden changes when estimating the escape direction in each frame independently, we generate a stable escape direction by computing the average between the current and the previous valid values. The robustness of the method is not affected despite the delay introduced by the filtering process.

2) Reactive Control Scheme: Given a valid and stable escape direction, the thruster setpoints are computed based on the position error \( e = (e_x, e_y, e_z) \) with respect to the center of the image \( P_c = (c_x, c_y) \).

\[
\begin{align*}
e_x &= \frac{D_{RoI}}{x_{RoI} - c_x} \quad e_y = \frac{y_{RoI} - c_y}{c_x} \quad e_z = -\frac{y_{RoI} - c_y}{c_y},
\end{align*}
\]

where \( D_{RoI} \) is the average depth in the selected RoI, and \( p_{RoI} = (x_{RoI}, y_{RoI}) \) is the escape direction on the reference frame image. Based on those references, we implemented a \( P \) controller for each degree of freedom of the OpenROV. The controllers are responsible for heave and surge motions and yaw rotation:

\[
\begin{align*}
u_s &= K_{ps} \cdot e_x, \\
u_y &= K_{py} \cdot e_y, \\
u_h &= K_{ph} \cdot e_z
\end{align*}
\]

where \( K_{ps}, K_{py} \) and \( K_{ph} \) are their proportional gains. In the actual implementation, the control signals are properly scaled to the range of the Electronic Speed Control (ESC) uses.

The output signal of the depth controller \( u_h \) is fed directly to the top thruster because it only has effect on the heave motion of the vehicle. The horizontal thrusters are driven by a combination of signals from \( u_s \) and \( u_y \) controllers. We added up these control signals, but assuming a different sign of \( u_y \) for each thrust [25].

III. EXPERIMENTAL RESULTS

We evaluated our algorithm in an offline sequence acquired in real oceanic environment and tested its online performance using a standard OpenROV [1] equipped with a single camera in a controlled environment. In both offline and online approaches, we compare our method against to the monocular obstacle avoidance proposed by Rodríguez-Telles et al. [12].

For the sake of a fair comparison, all methods were implemented using standard C++ with OpenCV [26] for efficient image processing, and sockets communication for a fast communication with the robot. We used two processing units: a notebook with an Intel I7-4510U@2.0GH\texttimes z CPU and 8Gb of RAM, and the standard OpenROV v2.7 onboard computer, a BeagleBone Black (BBB) with a Cortex A8@1GH\texttimes z CPU and 512Mb of RAM. The results performed using the
notbook requires the BBB to acquire the images and to transmit them using the tether.

[12] was implemented according to the paper, using a superpixel segmentation algorithm based on a modified version of the Simple Linear Iterative Clustering algorithm (SLIC) [13]. This modified SLIC was coded based on an open source project\(^1\). We also implemented the training step following the offline approach proposed by the authors in which the user indicates in some training images the superpixels corresponding to RoI.

Although all the evaluated sequences were acquired at 720p resolution, we re-scaled the images to \(640 \times 360\) pixels to achieve real-time performance in the control step \((\geq 10Hz)\). This resolution was enough to maintain the robustness of our system running at high frame rate.

A. Offline Experiments

We carried out offline experiments using a real oceanic sequence obtained with a Seabotix LBV300-5 ROV, equipped with a GoPro Hero3+ Black Edition camera. The image were acquired from a coral reefs with a sandy seabed in Brazil’s Northeast Coastal area at 10m water depth approximately. The video sequence shows challenging conditions such as floating sediment, fishes moving and illumination variation like sun flicker in a narrow passage scene.

Fig. 3 shows the offline experiments results for some key frames with the challenging situations stated before in the first row, figs. 3a-3d. The results of our method are depicted in the second row, figs. 3e-3h, and the results obtained with Rodríguez-Telles et al. [12] method are in the last row, figs. 3i-3l.

For our approach, we show the estimated depth map in gray scale, with the fitted ellipse (in cyan) and the escape direction (in yellow). Although the illumination changes in the scene, the RoI size is similar in all images since the adaptive threshold is based on the global light \(A\) value. As stated before, it is estimated by finding the intensity of the brightest pixel in the underwater dark channel (Eq. 5), and it changes accordingly the scene illumination.

The results obtained with [12] depicts the superpixel segmentation and their classification. Blue dots indicate superpixels classified as RoI and red dots represent the obstacles. The escape direction is also shown as a yellow circle. In all images, the method had some difficulty to discriminate between free and occupied areas, specially on the coral reef on the right side, in which many superpixels are shown as free. This causes the estimation of the escape direction to be unsafe in figs. 3j and 3l.

Table I shows the algorithms running time. Based on the current implementation, our algorithm is \(\times 25\) faster than [12]. Our algorithm can run up to 30Hz, while we could only achieve \(\sim 1.3Hz\) with [12], which is not enough for the control loop. Their performance in the execution time is highly dependent of the superpixel segmentation method \((\sim 90\%)\), whereas our algorithm is limited by the erosion operation using a circular kernel that is responsible to obstacle extent, which takes \(\sim 35\%\) of the time. The performance of our method on the BBB board is still limited to \(\sim 1Hz\).

B. Online Experiments

We also performed an online evaluation of the proposed algorithm with a OpenROV v2.7 [1], a small tethered ROV with three thrusters: two in horizontal plane for surge motion and yaw rotation, and a vertical one for heave motion. The vehicle is equipped with a Genius KYE F100 Ultra-wide angle full HD webcam. We assembled our standard unit without the laser pointers due to safety regulations. The OpenROV offers as low-cost alternative to traditional ROV platforms when operating in shallow water with no water currents.

We obtained the experimental results in a small circular pool with 1.5m of radius and 0.5m of water depth. Several marking cones were used as obstacles (see Fig. 4). The small water depth limited the escape direction that the method is able to compute. Therefore, we noted that the method had the tendency to move the ROV to the surface in those experiments. We reduced the proportional gain in the heave motion controller to compensate this effect.

Fig. 5 shows the results of our algorithm and [12] method during three different experiments. Our approach (figs. 5g-5l) computed the depth map and estimated a valid escape direction. The depth map estimation was not accurate because of the white floor. This is due to the limitation of the statistical prior that assumes darkness in the image in at least one color channel. The water surface can misclassify a free space due to reflection that generates a mirror effect of the white floor of the pool. Therefore, the method tried to compensate it with increasing the pitch of the vehicle. The results of the online experiments are shown on the attached video.

Our implementation of [12] was unable to correctly detect free and occupied areas. The obstacles were successfully detected only for images where the obstacles are in the center and near the camera, as well as in some areas on the vicinity of the center (figs. 5m, 5o, and 5r). Despite the correct detection in these cases, the algorithm was not able to compute a valid escape direction (highlighted with a red circle). Due to its limited capability to identify occupied area correctly, the method estimated the escape direction in the center of the image, i.e. the center of mass of the RoI as proposed in [12] and some collisions were incorrectly detected as a valid escape direction, e.g. Fig. 5n.

\(^1\)https://github.com/PSMM/SLIC-Superpixels
Fig. 3: Offline results for the real oceanic sequence: a-d) samples of the collected frames; e-h) results of our method showing the depth map, the fitted ellipse in the selected RoI and the escape direction; i-l) results of our implementation of Rodrigues-Telles et al. method with blue dots indicate superpixels classified as free areas, red dots represent obstacles and the escape direction.

Fig. 4: Experimental setup: an OpenROV platform in a pool where we conducted field tests. Obstacles were introduced (red cones) to evaluate the performance of the algorithms.

TABLE II: Comparative analysis of the proposed method against the state-of-the-art in terms of running time by frame for the online experiments.

<table>
<thead>
<tr>
<th>Method</th>
<th>Average Time (s)</th>
<th>Std. Deviation (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rodrigues-Telles et al. [12]</td>
<td>0.7698</td>
<td>0.0141</td>
</tr>
<tr>
<td>Our Method - Notebook</td>
<td>0.0396</td>
<td>0.0081</td>
</tr>
</tbody>
</table>

Table II shows the running time for the algorithms in the online experiments. Similarly to the offline case, our algorithm was ×19 faster than [12], with a smaller standard deviation. The execution time difference of our method with respect to the offline experiment is due to the size of the RoIs which imposes an increase of processing power requirements.

IV. CONCLUSIONS AND FUTURE WORK

This paper proposed a novel obstacle avoidance method for underwater environments using a single monocular camera. For each incoming frame and no previous information about the environment, our approach computed an estimation of the depth map respect to the camera using statistical priors and a physical underwater light attenuation model. After identifying the free areas on the depth map with an adaptive threshold, a fast segmentation method estimated the most promising RoI and computed the escape direction. This was turned into a reactive control action to avoid obstacles. We compared our approach against a state-of-the-art method in an offline dataset taken in a natural environment and in online experiments using an OpenROV platform in a controlled environment.

Future work will be focused on evaluating the accuracy of the depth map estimation under different illumination and water conditions. We will also explore the depth information to find an adaptive radius for a safer escape direction and to provide a multi-object segmentation. We will also install laser pointers or a simple sonar-based range finder to turn the depth map into actual distances. Furthermore, we will improve the code to make it able to run in real-time on the OpenROV onboard computer.

ACKNOWLEDGMENTS

We thank the colleagues from Autonomous Systems Laboratory, at CSIRO for hosting Paulo Drews-Jr during...
his sandwich program (sponsored by CAPES grant no. 999999.003584/2014-03) both for the prolific discussions and for their kind support in providing equipment and the necessary infrastructure for some of the experiments in this work. We also thank to VeRLab-UFMG and NAUTEC-FURG for providing equipment and assistance with part of the experimental data. This research is also partly supported by CNPq, CAPES and FAPEMIG.

## References


