# Obstacle Detection for Image-Guided Surface Water Navigation

Tanmana Sadhu*, Alexandra Branzan Albu*, Maia Hoeberechts‡, Eduard Wisernig†,
Brian Wyvill†
*Dept. of Electrical and Computer Engineering
†Dept. of Computer Science
‡Ocean Networks Canada
University of Victoria, Victoria, BC, Canada

***Abstract*—Maritime safety is an issue of concern when operating a small to medium sized sailboat, and the appearance of hazards in the navigational route like floating logs can lead to a severe collision if undetected. As a precautionary measure to prevent such a collision with a log, a 2D vision-based detection system algorithm presented to detect these floating hazards. We take a combined approach involving predictive mapping by linear regression and saliency detection. The proposed method has been evaluated using precision and recall measures. The evaluation results show that the algorithm is robust and computationally non-intensive for future implementation of a real time on-board obstacle detection system for autonomous and computer-assisted sailboat navigation.**

***Index Terms*—obstacle detection; maritime navigation; linear regression; saliency;predictive mapping**

## I. Introduction

Despite significant progress in autonomous navigation for air and land vehicles, the area of autonomous water navigation is still relatively unexplored. Navigational aids in technology such as GPS localization are currently in practical use, however, information regarding obstacles on the water surface which may appear unexpectedly, has to be largely obtained by visual means. At present obstacle avoidance, area survey analysis, threat assessment and surveillance is dependent on man-in-the-loop situation awareness [1]. As noted in a US Coast Guard report from 2014 [2], accidents such as collision with floating debris during the operation of a sailboat can commonly occur due to attention deficit or misjudgement due to inexperience []

Safety is a primary concern for an operator during the operation of a sailing vessel, and hence the need to understand the surrounding environment, and awareness about any hazards on the water surface is crucial. On-the-fly detection of obstacles and subsequent issuance of warning using only a monocular camera device and limited on-board processing would offer a significant advantage in such a case. On sailing around shores and inlets, floating debris, especially logs which are half-submerged, are prone to appearing unexpectedly, often requiring the operator to maneuver the boat in time to prevent a collision. Very few attempts have been made to implement a system for water-based navigation which may generate a warning on an impending collision situation with the aid of computer vision technology. The task is complicated by the visually noisy, unstructured scene consisting of the shoreline, reflections and the rapidly changing water features. Widely varying illumination conditions also make it difficult to differentiate between obstacles and other floating objects which do not qualify as obstacles (birds, seaweed, etc.).

We present an approach to detect floating debris, specifically logs, which might be hazardous obstacles in the course of a small sailing vessel, as a step towards solving the debris detection and alarm generation problem. We take a modular approach based on classification by generalized linear regression and saliency-based techniques in order to generate a combined probability map, which is further thresholded and analyzed by post-processing steps to perform the detection. We show that the combination of the regression-based predictive map and contrast-based saliency map for the goal of obstacle detection on our locally acquired database yield good detection results. Another contribution of the work is the development of an image/video database consisting of high resolution frames that can be used to study obstacle detection and avoidance for water navigation in a more comprehensive manner.

The rest of the paper is divided into sections as follows: Related work is presented in Section 2. The proposed algorithm is described in Section 3. Experimental results are discussed in Section 4, and Section 5 discusses the conclusions and future directions.

## II. Related Work

The task of obstacle detection and tracking has been extensively researched for terrestrial autonomous mobile systems. With the increased availability of computing resources, vision research for such systems has moved outside of controlled experimental and factory setup [3]. Road-based autonomous vehicular systems for terrestrial navigation have been researched in [4], [5], [6], [7].The problem of path planning for a robotic land vehicle, for instance, is addressed [8] in which trajectories are found by a combination of 2D visual detection and 3D mapping based on stereo depth. The authors in [9] on the other hand, approach the obstacle detection problem using 2D monocular vision only, by an appearance-based method with online learning of color histograms. However, while the simplicity of such a method allows it to perform well in real time, we need to consider the complexity of our background

978-1-5090-2491-9/16 $31.00 © 2016 IEEE
DOI 10.1109/CRV.2016.34

scenarios, as the sailboat navigates close to shores in vastly different natural lighting conditions.

A large majority of the recently proposed algorithms rely on some machine learning-based techniques to produce accurate results. The authors in [10] apply a reinforcement learning technique for measuring relative depth using only monocular visual cues on images of outdoor environments. This algorithm uses texture energies and texture gradients to compute feature vectors over image stripes. The output of the vision system is then converted into steering commands by reinforcement learning, enabling the conversion of the learning output to robot navigation in real time.

Object detection in marine environments has been attempted in [11], [12] and [13]. The method in [13] aims to find objects of interest in still images of the sea surface using adaptive thresholding on texture properties. This method fails to operate well in our database as it cannot always account for widely varying texture patterns in different parts of the water surface, and an accurate segmentation of the object of interest against the water surface in this way is unrealistic. The authors in [11] focus upon the detection of marine vehicles in open sea environments. The method is however, dependent on horizon detection as a primary step of the algorithm, to delineate water from sky prior to object detection. Our challenge with robust horizon detection [14] is that the horizon line is not always clearly defined as a sky-water separation, as the visual scene almost always contains islands and shores that break the continuity of the horizon lines.

The method in [13] is different from the above-described methods as it operates on IR images for locating floating mines. The authors have compared a number of algorithms for detection on an agitated sea surface by background subtraction and motion-based methods on IR imagery. Background subtraction as a problem of saliency has been presented in [15] and the model has been demonstrated to work well with moving target objects against a dynamic background. Saliency has been recognized as an important cue for visual obstacle avoidance in indoor environments in many works ( [16], [17], [18]). The authors in [19] proposed a saliency model which is an adaptation for obstacle detection in all-terrain environments of the biologically inspired model originally proposed by [20]. Their method is based on the idea that the higher the saliency value of a given region, the more likely it is for the visual field to contain an object of significance.

Research for visual obstacle avoidance within a context similar to ours has been presented in [21] for riverine environment navigation that uses an optical-flow-based system. While their methods reportedly yield good results for obstacle detection, they return estimates of the regions of all possible obstacles present in a frame as opposed to specific targets.

Moreover, our target objects do not always have a strikingly different motion pattern, especially in calm waters and when located further away from the camera. Hence the motion features are not found to be informative enough to apply optical-flow-based methods. Instead we take up a regression-based predictive mapping approach to create a probability map for our target class. We also employ saliency detection which has been previously applied in terrain-based navigation systems to reinforce our target detection. We propose a novel way of combining these two main steps of the algorithm for our application. This approach is designed keeping in mind feasibility for a future real time implementation with an inexpensive on-board processing unit in terms of computational load, wherein the detection system output will be directed to a sailboat maneuvering system to generate alerts.

## III. Proposed Method

In order to achieve real-time performance with limited hardware resource availability, we choose an algorithmic approach that would not be highly computationally intensive and yet have a reasonably high degree of robustness.

To achieve improved performance, a combination of techniques is used in our approach, which are applied in the sequence shown in the overview block diagram (see fig. 1). A linear regression model (explained below in section III. (C)) is implemented as the predictive mapping by regression analysis step. The saliency detection module (section III. (D)), applied in combination with regression allows for a dramatic reduction in the number of false positives, which occur mainly due to the presence of the shoreline in many frames, along with reflections and light patterns on the water surface. In experiments, detection by thresholding the regression map only yielded many false positives, and thresholding only the saliency map led to increased false negatives, motivating the use of a combined approach. Though some methods have been reported which make use of saliency along with other methods for detection ( [22], [23]) the combination of linear regression analysis with saliency for obstacle detection has not been, to the best of our knowledge, been implemented before.

The steps of the proposed algorithm are described in detail below.

### A. Preprocessing:

This step is performed to reduce noise effects prior to computing of the feature values. Each image frame is convolved with a rotationally symmetric 2-D Gaussian filter of size $3 \times 3$ which has an overall smoothing effect. A standard deviation value $\sigma = 0.5$ for the kernel was empirically found to to be optimal taking into consideration the smallest size of the object to be detected (300 pixels approximately).

### B. Feature Extraction:

The features are extracted from each image on a pixel-by-pixel basis for classification. We consider appearance-based features and group them into two different categories: (1) intensity and color-based and (2) texture-based. The first category comprises one intensity channel ($I$) and two color channels ($RG$,$BY$), and the second comprises the texture features as described in [20]. The intensity ($I$), red/green ($RG$) and blue/yellow ($BY$) channels are computed as in eqns. (1-3) below [24]:

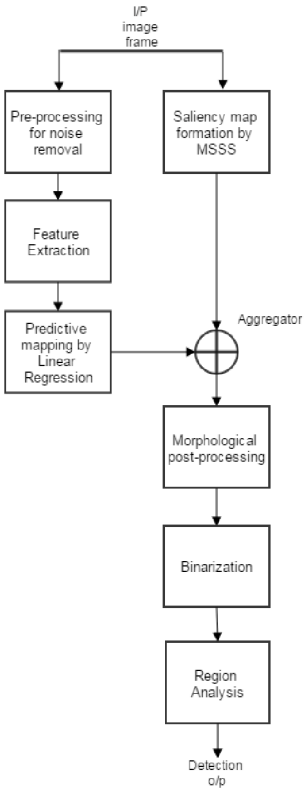



Fig. 1: Block diagram

$$I = r + g + b \tag{1}$$

$$RG = r - g \tag{2}$$

$$BY = b - \frac{r+g}{2} - \frac{min(r,g)}{2} \tag{3}$$

The texture of water surface regions are significantly different from other regions (for instance ripples on the water surface versus logs). Computation of the statistical moments of the gray level histogram of the image is used to model these variations in order to differentiate between textures of the logs and the background or other objects not of interest. The entropy is one such texture measure, used to measure the randomness in intensity distribution of an image region. It is represented in the eqn. (4) below [25]:

$$E = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)log(p(i,j)) \tag{4}$$

where $E$ denotes entropy, $p(i,j)$ is the $(i,j)th$ entry in the normalized gray tone spatial dependence matrix $P(i,j)$, and $N_g$ is the number of gray levels in the quantized image. The entropy feature value is calculated for each pixel in the image frame by considering a $5 \times 5$ window.

The other set of texture features computed are the Local Binary Pattern (LBP) [26] values at each radial pixel neighborhood. The LBP measure encodes the relationship between a pixel and its neighbors into a binary word, allowing the representation of patterns. It can be represented by the following equation:

$$LBP(x_i, y_j) = \sum_{n=0}^{N-1} f(g_n - g_{i,j})2^n \tag{5}$$

where $g_{i,j}$ is the gray value of center pixel at $(i,j)$ location in a circular neighborhood,$g_n$ is the gray value of its neighbor. The function $f$ is defined as $f(x) = \begin{Bmatrix} 1 & x \geq 0 \\ 0 & x < 0 \end{Bmatrix}$

The histograms computed over each pixel neighborhood are used as texture descriptors.

### *C. Predictive mapping by Linear Regression Model:*

In this step we apply a supervised learning technique based on the linear regression model. Regression analysis involves fitting a model to data and has been used previously in computer vision for a variety of applications such as face detection and land-based autonomous vehicular navigation [27], [28], [10]. The goal of application of regression at this step of our algorithm is to obtain a map which assigns a probability measure to each pixel in the image that predicts how likely it is for the pixel to belong to target object. Based on the observation that features from our target class (logs) form patterns that can be mapped to the same linear subspace, we apply the classical linear regression technique for class probability prediction, which involves linearly combining the input feature vectors [29]. To estimate the coefficients of the regression model, we consider the number of classes $L = 2$, corresponding to target and background, while $m$ is the number of training images. We model the training set as:

$$S_{(a\times b)} = \bigcup_i q_i^{(c)}, \qquad c = 1..L, i = 1..m \tag{6}$$

where $q_i$ represents each training image of the order $a \times b.$, and $i = 1...m$.

The set of all feature vectors can be represented as:

$$W_{(g\times h)} = \bigcup_i \mathbf{w}_i^{(c)}, \qquad c = 1..L, i = 1..m \tag{7}$$

where each feature vector $\mathbf{w}_i$ (corresponding to each training image) is of the order $g \times h$, and $g = a*b$ and $h$ is the number of features.

The training image set is transformed to the feature vector set as $S_{(a\times b)} \rightarrow W_{(g\times h)}$ by concatenation of each feature vector $\mathbf{w}_i$.

The feature vectors are further normalized and stacked together to form the vector $X_c$ as follows:

$$X_c = [\mathbf{w}_1^{(c)}, \mathbf{w}_2^{(c)}, ....\mathbf{w}_m^{(c)}]^T, \qquad c = 1..L \tag{8}$$

Each vector $X_c$ is specific to a class $c$ , and is also called the regressor for the class. Let our class-specific dependent target variable be denoted by $Y_c$, given by a deterministic output function $y(X_c), \beta_c$ where $\beta_c$ is the optimal set of parameters. This parameter set is estimated through the training phase as $\tilde{\beta}$ and is used to model the linear dependency between the independent vector of regressors $X_c$ and scalar dependent target variable $Y_c$. The eqn.(9) below shows the relationship between dependent target and independent regressor vectors:

$$Y_c = X_c\beta_c + \epsilon, \qquad c = 1...L. \tag{9}$$

where a disturbance term $\epsilon$ represents an unobserved random variable, which adds noise to the linear model. The estimated model parameter vector $\beta_c$ can be estimated by the least squares estimation method [30]:

$$\tilde{\beta_c} = (X_c^T X_c)X_c^T Y_c \tag{10}$$

In the least squares method the sum of squares of errors error function $E_D$ is minimized as follows:

$$E_D = \frac{1}{2}\left\|Y_c - \tilde{\beta_c}\left(X_c\right)\right\|^2 \tag{11}$$

Next we construct the test feature vector $Z$ corresponding to each test image of the order $g \times h$, similar to the training phase.

The probability measure of the test feature vectors to belong to a class $c = 1...L$ is calculated using the above linear regression model as follows:

$$R = P(Z(c = 1)|\tilde{\beta}) \tag{12}$$

Here $c = 1$ corresponds to the target class and $R$ is the regression map which we obtain corresponding to each test image frame, represented by a visualization in fig. 2b.

### *D. Saliency detection using Maximum Symmetric Surround Saliency (MSSS):*

Visual saliency is the perceived quality that may be attributed to the most distinct region in an image, with respect to color, brightness or some other characteristics, as compared to other regions. A pixel is usually considered salient if it belongs to an area of image that has a unique appearance, both locally and globally [31]. From observation of our video database, the logs often appear highly contrasted to the surrounding water regions in our frames, even in varying illumination. This observation led us to consider applying a saliency detection algorithm that would be fast to compute and improve the detection capability of the algorithm (by eliminating false positive occurrences) from the previous step. We chose to implement the Maximum Symmetric Surround Saliency technique proposed in [32] which proved to be relatively fast and accurate for our detection task. It is based on the frequency tuned center-surround saliency measure introduced in [33]. The method computes the center-surround saliency measure for each pixel in CIELAB color space, which includes a wide range of perceivable colors. Their algorithm considers the surround as a symmetric region around the center whose dimension decreases as we move the center closer to the border of the image.

For an image of width $w$ and height $h$, the symmetric surround saliency value at the given pixel location $(r, t)$ is obtained as:

$$S_{MSSS}(r, t) = \left\|\tilde{I}(r, t) - I_{lp}(r, t))\right\| \tag{13}$$

where $\tilde{I}(r, t)$ is the average value of CIELAB vector for the subimage with center pixel at location (r,t) given by eqn. (14) as follows:

$$\tilde{I}(r, t) = \frac{1}{W}\sum_{m=r-r_0}^{r+r_0}\sum_{n=t-t_0}^{t+t_0} I(m, n) \tag{14}$$

where $r_0 = min(r, w - r)$; $t_0 = min(t, h - t)$; $W = (2r_0 + 1)(2t_0 + 1)$;

### *E. Aggregation:*

The predicted output map from (C) and the saliency map from (D) are normalized such that all the values lie within the same range between [0,1]. They are then combined into a final map using an aggregation technique which assigns a weight $\mu$ to the regression map output and a weight of $\lambda$ to the saliency map output in the following manner:

$$F = \mu * R + \lambda * S_{MSSS} \tag{15}$$

where $F$ represents the final probability map. The weights $\mu$ and $\lambda$ are set to 0.7 and 0.3 respectively. These parameter values are determined by computing the precision recall values at each different combination of $\mu$ and $\lambda$ for each test set. The resulting curve is shown in fig. 3. We chose the values for which the precision and recall values are the highest (see fig. 3).

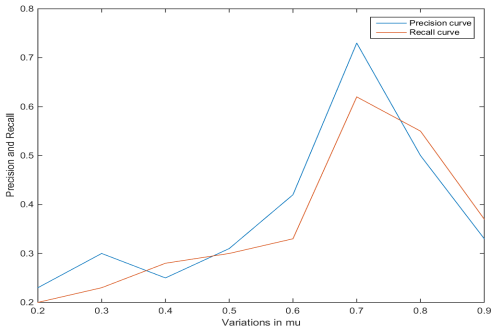
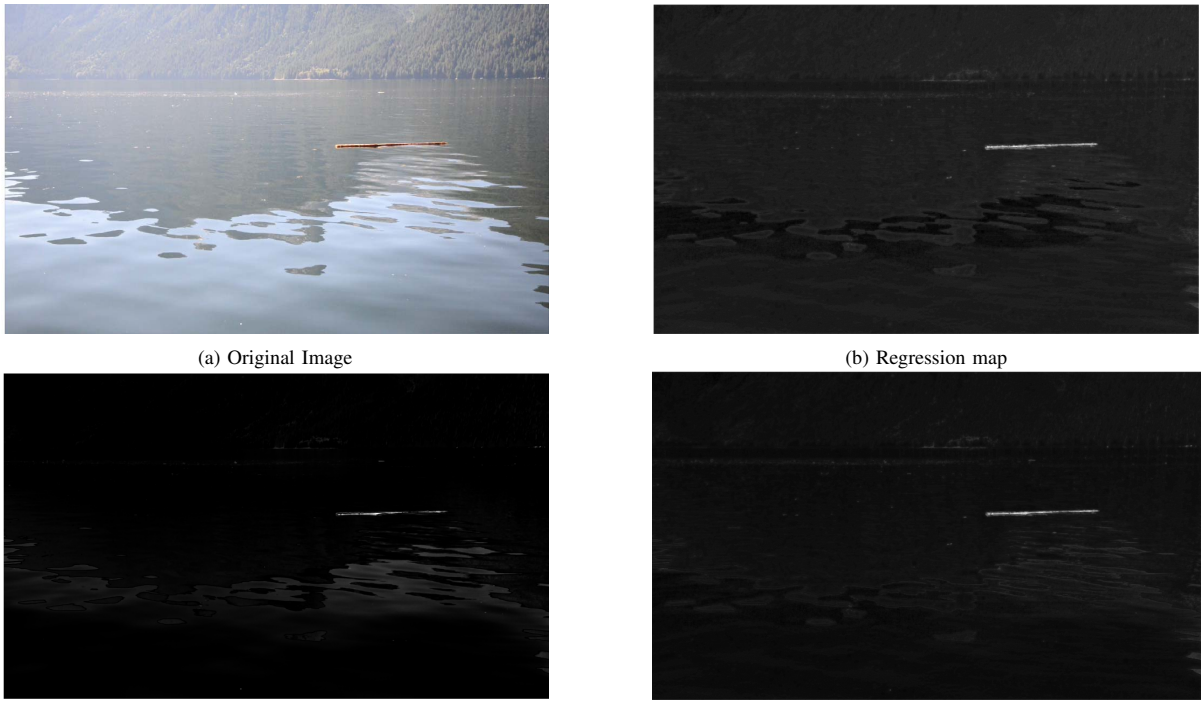



Fig. 3: Comparison of the different parameter values for aggregator

### *F. Post-processing and shape analysis*

The final combined map $F$ obtained from the above step is normalized and thresholded to obtain a binary map with non-target pixels set to 0 (black) and target region pixels set to 255 (white). We use a threshold of 70% of the maximum value of



(a) Original Image (b) Regression map

(c) Saliency map (d) Final map

Fig. 2: Comparison between the (a) original image, (b) regression map, (c) saliency map and (d) final combined map.

$F$ on the basis of observation of the entire test dataset, which can be varied by about 10% of its value without significantly affecting the final segmentation. The binary map thus obtained contains many noisy connected component regions which need to be removed through the post-processing steps, which are described below.

a. Morphological filling and closing - This is performed on the binary map to fill holes and remove noise elements. We use a circular structuring element of dimension 8 to perform this step.

b. Removal of connected components associated with the image border - This, in some cases, helps to exclude shore components from the binary map.

We further impose constraints on each of the following region properties, which can be considered as parameters, and their values have been empirically determined by experimenting over the entire test dataset. They are as follows:

a) The area of the region ($A$) in pixels is in the range $300 \leq A \leq 40000$ to approximately limit the area between the smallest and largest pixel dimensions, respectively, of the target objects in our database.

b) Eccentricity $e = \frac{Majoraxislength}{Minoraxislength}$, is a measure of the deviation of shape from circularity such that for a circle $e = 0$. We restrict this measure as $0.9 \leq e \leq 1$ based on the rationale that values closer to 1 would indicate long, thin objects likely to be logs.

c) Solidity($s$) is the measure of the convexity (or concavity) of a shape defined by $s = A_s/H$ where $A_s$ represents the area of the shape and $H$ is the convex hull area. Restricting its value range as $0.5 \leq s \leq 1$ helps to remove component regions with very rough edges, which are unlikely to be our targets.

d) Euler number ($EN$) indicates the difference between number of contiguous parts ($P$) and number of hole regions ($D$) such that $EN = P - D$. Its value in our case must be 1 as the logs to be detected generally do not have large hole regions.

The final objective of the detection system is to generate an alert once an obstacle is located. To achieve that, in our algorithm we evaluate a series of 10 successive frames at a time, over a time frame of roughly 20 seconds given a computational time of 2.6 seconds per frame on average. If an obstacle is detected over all the frames, an alert flag is set to be high.

## IV. Evaluation and Results

### A. Experimental database

The data used in this research consists of videos and images acquired using a forward looking monocular camera (Canon 7D, using a 15-85mm Canon stabilized lens) from the bow of a sailing vessel overlooking the water surface. The experimental database comprises a set of 100 training frames and 1400 test

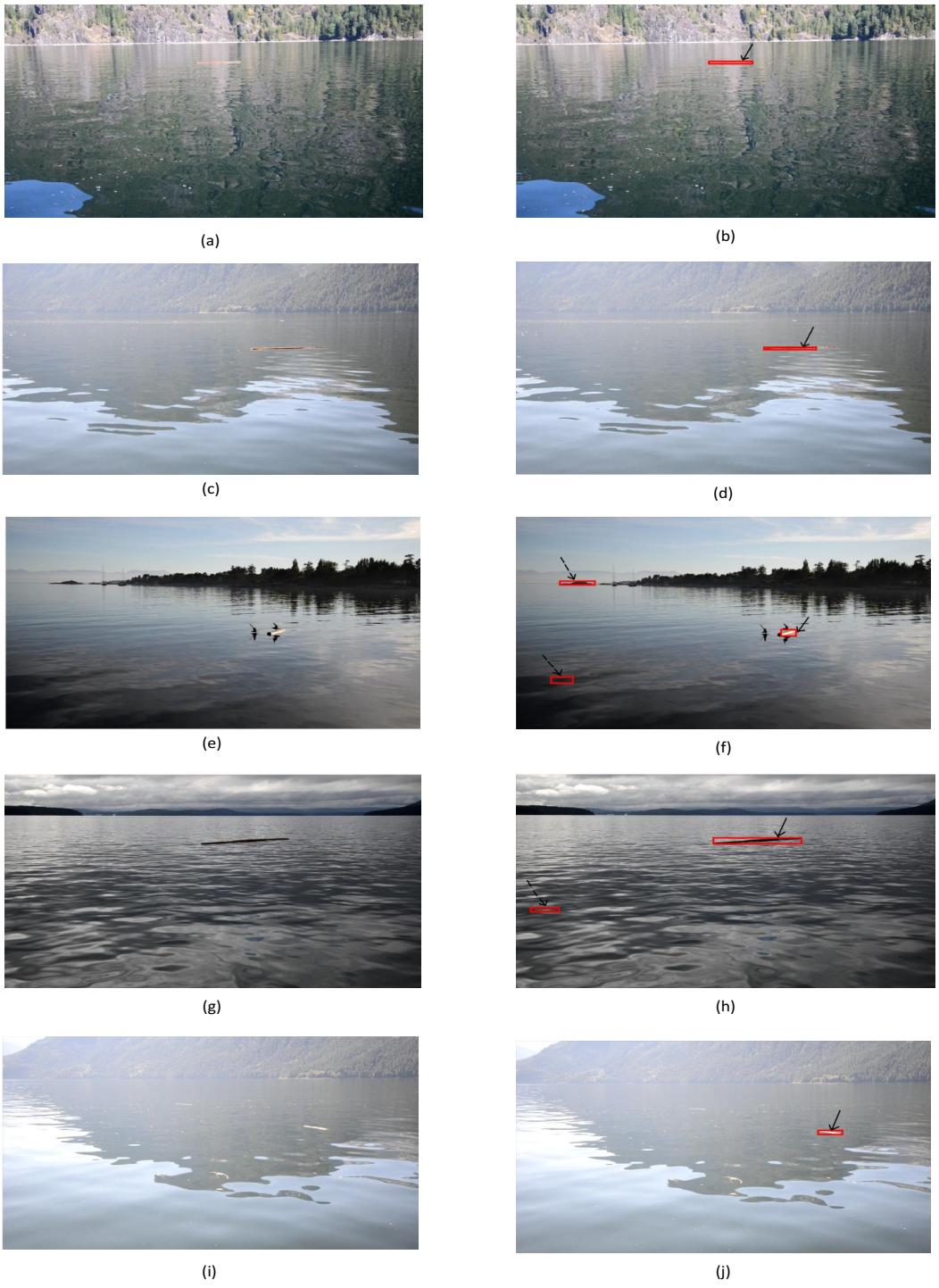



Fig. 4: Left hand column shows original image frame, right hand column shows detection results. (a),(b)Set 1 consists of frames where the shore is strongly reflected. (c),(d)Set 2 image frame has large salient patches (e),(f) Set 3 frames with strong reflections and varying illumination (g),(h) Set 4 with cloudy conditions (i),(j)Set 5 frame has small-sized obstacle further away from camera, thus hard to detect. [Correct detections shown by solid line arrows, false detections shown by broken line arrows.]

frames with high spatial resolution of $1920 \times 1080$, obtained from 4 different videos sequences of at least 3 minutes length each.

### B. Evaluation

Performance of the proposed method is highly dependent on the characteristics of the test images. For this reason, the test frames were divided into 5 different sets of 200-300 frames each, with each set representing different background and/or illumination conditions. Set 1 (see fig. 4) consists of frames which show a log present at close proximity to the shore. The water almost entirely reflects the shore. The appearance of water in set 2 varies considerably within the frame, with large salient patches. Set 3 has frames with prominent shoreline and varying illumination. The horizon is difficult to identify in many images in set 3. Set 4 is distinguished by its illumination owing to cloudy conditions. The log in set 5 is further away from the camera and thus hard to detect. Graphical examples of detection results (marked by red bounding boxes) are shown in fig. 4. Correct detection results are highlighted by solid line arrows while false detections are shown with broken line arrows. In the example images, the logs are detected accurately in all of the sample images, however, fig. 2. (f) and (h) show some detected false positives caused by ripples in water and part of the shore.

For the purpose of evaluation, we compare our proposed method to two other standard supervised methods for predictive mapping: backpropagation neural networks (NN) and Gaussian Mixture Models (GMM). The saliency detection step (MSSS) was applied consistently across all three processing scenarios. The goal of this evaluation is to determine the best predictive method to use in conjunction with the saliency detection technique to achieve better accuracy and fewer false alarms.

We use the backpropagation neural network(NN) [34] to find a predicted probability map similar to the regression map $R$ calculated in the previous section combined with the MSSS (optimal no. of hidden layers = 3, $\mu = 0.7$,$\lambda = 0.3$). The Gaussian Mixture Model (GMM) where model parameters are estimated by Maximum Likelihood Estimation [35] is similarly used in combination with the MSSS (maximum no. of iterations=100,$\mu = 0.7$,$\lambda = 0.3$).

The test sets have been evaluated by precision ($P$) and recall ($R$) measures (eqns. 16,17). The $P$ and $R$ measures are given by :

$$Precision(P) = \frac{(TP)}{(TP) + (FP)} \tag{16}$$

$$Recall(R) = \frac{(TP)}{(TP) + (FN)} \tag{17}$$

where $TP$ and $FP$ denote True Positives and False Positives respectively, while $FN$ denotes False Negatives.

Table I presents the evaluation results for our method in comparison to other supervised methods for predictive mapping, keeping the saliency detection step (MSSS) the same.

TABLE I: Experimental Results

| Set | Proposed method | | GMM+MSSS | | NN+MSSS | |
|---|---|---|---|---|---|---|
| | Precision | Recall | Precision | Recall | Precision | R |
| 1 | 91.6% | 95.0% | 80.8% | 98.6% | 71.2% | 87.6% |
| 2 | 97.0% | 85.6% | 91% | 93.2% | 54.8% | 91.0% |
| 3 | 29.7% | 98.6% | 8.1% | 88.3% | 8.6% | 100.0% |
| 4 | 75.9% | 100.0% | 82.4% | 100.0% | 98.6% | 100.0% |
| 5 | 87.6% | 90.0% | 83.5% | 90.0% | 80.5% | 65.4% |

The average computational time for our algorithm is calculated to be 2.6 seconds per frame on a 1.8 GHz Intel Core i5-3337 processor using MATLAB®version 8.5. The results show that the precision and recall values for the proposed method are higher, for most cases, in comparison to the other methods for the test dataset. For set 3, although the log is detected accurately, the number of false positives is high, causing the precision and recall to be drastically lower than for the other sets. This occurs due to the detection of a part of the shore similar in appearance to the logs (see fig. 4 (e)). This limitation may be addressed in a future implementation by accurately delineating water region from shore.

## V. Conclusion

We present a novel method of applying linear regression analysis in conjunction with the Maximum Symmetric Surround Saliency detection algorithm for target object detection. The results show that the linear regression predictive mapping has a good success rate in comparison to other supervised techniques for our application. Our algorithm uses low level features of intensity color and texture, and performs accurate detections with low computational complexity. We show that our method is relevant for the application of log detection and could be extended to obstacle detection in general for future applications. In its current implementation, our method performs only alert generation through object detection. A tracking module can be integrated into the system in a future implementation, to detect the exact position of the obstacle at each frame. For a real time implementation, using a GPU for performing some of the image processing tasks might significantly improve computational efficiency and reduce computational times. We could also consider eliminating the shoreline from the frames before the detection process by integrating location information from standard terrestrial charts.

## References


[1] F. D. Snyder, D. D. Morris, P. H. Haley, R. T. Collins, and A. M. Okerholm, "Autonomous river navigation," in *Optics East*. International Society for Optics and Photonics, 2004, pp. 221–232.

[2] R. B. Matthew D. Plumlee, Roland Arsenault and C. Ware, "Chart of the future project: Maximizing mariner effectiveness through fusion of marine & visualization technologies," Washington, DC, November 16 - November 17 2004. [Online]. Available: www.chc2012.ca

[3] L. Fletcher, N. Apostoloff, J. Chen, and A. Zelinsky, "Computer vision for vehicle monitoring and control," in *Proc. Australian Conference on Robotics and Automation*, 2001.

[4] V. Graefe and K.-D. Kuhnert, "Vision-based autonomous road vehicles," in *Vision-based Vehicle Guidance*. Springer, 1992, pp. 1–29.

[5] Y. He, H. Wang, and B. Zhang, "Color-based road detection in urban traffic scenes," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 5, no. 4, pp. 309–318, 2004.

[6] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouveliere, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 3, pp. 589–606, 2010.

[7] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-based intelligent vehicles: State of the art and perspectives," *Robotics and Autonomous systems*, vol. 32, no. 1, pp. 1–16, 2000.

[8] K. Yamaguchi, T. Kato, and Y. Ninomiya, "Moving obstacle detection using monocular vision," in *Intelligent Vehicles Symposium, 2006 IEEE*. IEEE, 2006, pp. 288–293.

[9] I. Ulrich and I. Nourbakhsh, "Appearance-based obstacle detection with monocular color vision," in *AAAI/IAAI*, 2000, pp. 866–871.

[10] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 593–600.

[11] S. Fefilatyev, "Detection of marine vehicles in images and video of open sea," Ph.D. dissertation, University of South Florida, 2008.

[12] P. Mulassano and L. L. Presti, "Object detection on the sea surface, based on texture analysis," in *Electronics, Circuits and Systems, 1999. Proceedings of ICECS'99. The 6th IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 855–858.

[13] A. Borghgraef, O. Barnich, F. Lapierre, M. Van Droogenbroeck, W. Philips, and M. Acheroy, "An evaluation of pixel-based methods for the detection of floating objects on the sea surface," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, p. 5, 2010.

[14] Z. Ji, Y. Su, J. Wang, and R. Hua, "Robust sea-sky-line detection based on horizontal projection and hough transformation," in *Image and Signal Processing, 2009. CISP'09. 2nd International Congress on*. IEEE, 2009, pp. 1–4.

[15] V. Mahadevan and N. Vasconcelos, "Background subtraction in highly dynamic scenes," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–6.

[16] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, and D. G. Lowe, "Curious george: An attentive semantic robot," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 503–511, 2008.

[17] J. Morén, A. UDE, A. Koene, and G. Cheng, "Biologically based top-down attention modulation for humanoid interactions," *International Journal of Humanoid Robotics*, vol. 5, no. 01, pp. 3–24, 2008.

[18] Y. Yu, G. K. Mann, and R. G. Gosine, "A task-driven object-based attention model for robots," in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*. IEEE, 2007, pp. 1751–1756.

[19] P. Santana, M. Guedes, L. Correia, and J. Barata, "A saliency-based solution for robust off-road obstacle detection," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3096–3101.

[20] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 11, pp. 1254–1259, 1998.

[21] T. El-Gaaly, C. Tomaszewski, A. Valada, P. Velagapudi, B. Kannan, and P. Scerri, "Visual obstacle avoidance for autonomous watercraft using smartphones," 2013.

[22] L.-K. Wong and K.-L. Low, "Saliency-enhanced image aesthetics class prediction," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 997–1000.

[23] J. Han, K. N. Ngan, M. Li, and H.-J. Zhang, "Unsupervised extraction of visual attention objects in color images," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, no. 1, pp. 141–145, 2006.

[24] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "Sun: A bayesian framework for saliency using natural statistics," *Journal of vision*, vol. 8, no. 7, p. 32, 2008.

[25] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[26] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[27] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 11, pp. 2106–2112, 2010.

[28] X. Chai, S. Shan, X. Chen, and W. Gao, "Locally linear regression for pose-invariant face recognition," *Image Processing, IEEE Transactions on*, vol. 16, no. 7, pp. 1716–1725, 2007.

[29] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[30] S. M. Stigler, "Gauss and the invention of least squares," *The Annals of Statistics*, pp. 465–474, 1981.

[31] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 10, pp. 1915–1926, 2012.

[32] R. Achanta and S. Süsstrunk, "Saliency detection using maximum symmetric surround," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 2653–2656.

[33] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on*. IEEE, 2009, pp. 1597–1604.

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[35] G. McLachlan and D. Peel, *Finite mixture models*. John Wiley & Sons, 2004.