A graphical model for rapid obstacle image-map estimation from unmanned surface vehicles

Matej Kristan[1,2], Janez Perš[1], Vildana Sulič[1], Stanislav Kovačič[1]

[1]Faculty of computer and information science University of Ljubljana, [2]Faculty of electrical engineering University of Ljubljana

Abstract. Obstacle detection plays an important role in unmanned surface vehicles (USV). Continuous detection from images taken onboard the vessel poses a particular challenge due to the diversity of the environment and the obstacle appearance. An obstacle may be a floating piece of wood, a scuba diver, a pier, or some other part of a shoreline. In this paper we tackle this problem by proposing a new graphical model that affords a fast and continuous obstacle image-map estimation from a single video stream captured onboard a USV. The model accounts for the semantic structure of marine environment as observed from USV by imposing weak structural constraints. A Markov random field framework is adopted and a highly efficient algorithm for simultaneous optimization of model parameters and segmentation mask estimation is derived. Our approach does not require computationally intensive extraction of texture features and runs faster than real-time. We also present a new, challenging, dataset for segmentation and obstacle detection in marine environments, which is the largest annotated dataset of its kind. Results on this dataset show that our model compares favorably in accuracy to the related approaches, requiring a fraction of computational effort.

1 Introduction

Obstacle detection is of central importance for lower-end small unmanned surface vehicles (USV) used for patrolling coastal waters (see Figure 1). Such vehicles are typically used in perimeter surveillance, in which the USV travels along a pre-planned path. To quickly and efficiently respond to the challenges from highly dynamic environment, the USV requires an onboard logic to observe the surrounding, detect potentially dangerous situations, and apply proper route modifications. An important feature of such vessel is the ability to detect an obstacle at sufficient distance and react by replanning its path to avoid collision. The primary type of obstacle in this case is the shoreline itself, which can be avoided to some extent (although not fully) by the use of detailed maps and the satellite navigation. Indeed, [1] proposed an approach that utilizes an overhead image of the area obtained from Google maps to construct a map of static obstacles. But such an approach cannot handle a more difficult class of dynamic obstacles that do not appear in the map (e.g., boats, buys and swimmers).

A small USV requires ability to detect near-by and distant obstacles. The detection should not be constrained to objects that stand out from the water, but

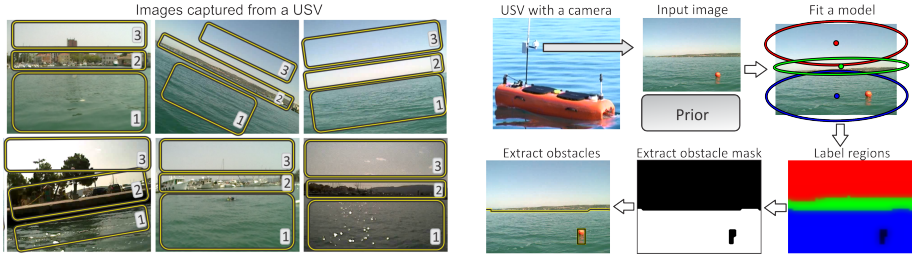

Fig. 1. Images captured from the USV split into three semantically different regions (left) and our approach for obstacle image-map estimation (right).

should also detect flat objects, like debris or emerging scuba divers, etc. Operation in shallow waters and marinas constrains the size of USV and prevents the use of additional stabilizers. This puts further constraints on the weight, power consumption, types of sensors and their placement. Cameras are therefore becoming attractive sensors for use in low-end USVs due to their cost-, weight- and power-efficiency and a large field of view coverage. This presents a challenge for development of highly efficient computer vision algorithms tailored for obstacle detection in a challenging environments that the small USVs face. In this paper we address this challenge by proposing a segmentation-based algorithm for obstacle-map estimation that is derived from optimizing a new well-defined graphical model and runs at over 70fps in Matlab on a single core machine.

1.1 Related work

The problem of obstacle detection has been explicitly or implicitly addressed previously in the field of unmanned ground vehicles (UGV). In a trail-following application [2] use an omnidirectional camera to detect trail as a region that is most contrasted to its surrounding, however, dynamic obstacles are not addressed. [3, 4] address the problem of low-proximity road detection of laser scanners by bootstrapping color segmentation with the laser output. The proximal road points are detected by laser, projected to camera and used to learn a Gaussian mixture model which is in turn used to segment the rest of the image captured by the camera. Combined with horizon detection [5], this approach significantly increases the distance at which the obstacles on the road can be detected. Alternatively, [6] casted the obstacle detection as a labelling task in which they employ a bank of pre-trained classifiers to 3D point clouds and a Markov random field to account for the spatial smoothness of the labelling.

Most UGV approaches for obstacle detection explicitly or implicitly rely on ground plane estimation from range sensors and are not directly applicable to aquatic environments encountered by USV. Scherer et al. [7] propose a water detection algorithm using a stereo bumblebee camera, IMU/GPS and rotating laser scanner for navigation on a river. Their system extracts color and texture features over blocks of pixels and eliminates the sky region using a pre-trained

classifier. A horizon line, obtained from the onboard IMU, is then projected into the image to obtain samples for learning a color distribution of the regions below and above horizon, respectively. Using these distributions, the image is segmented and results of the segmentation are used in turn, after additional postprocessing steps, to train a classifier. The trained classifier is fused with a classifier from the previous frames and applied to the blocks of pixels to detect the water region. This system relies heavily on the quality of hardware-based horizon estimation, accuracy of pre-trained sky detector and the postprocessing steps. The authors report that the vision-based segmentation is not processed onboard, but requires special computing hardware, which makes it below a realtime segmentation at constrained processing power typical for small USVs.

Some of the standard range sensor modalities for autonomous navigation in maritime environments include radar [8], sonar [9] and ladar [10]. Range scanners are known to poorly discriminate between water and land in the far field [11], suffer from angular resolution and scanning rate limitations, and poorly perform when the beam's incidence angle is not oblique with respect to the water surface [12, 13]. Several researchers have thus resorted to cameras [14, 15, 10, 16, 17, 13] for obstacle and moving object detection instead. To detect dynamic objects in harbor, [14] assume a static camera and apply background subtraction combined with motion cues. However, background subtraction cannot be applied to a highly dynamic scenes encountered on a moving USV. [17] attempt to address this issue using stereo systems, but require large baseline rigs that are less appropriate for small vessels due to increased instability and limit processing of near-field regions. Santana et al. [13] apply fusion of Lukas Kanade local trackers with color oversegmentation and a sequence of k-means clusterings on texture features to detect water regions in videos. Alternatively, [15, 16] apply a low-power solution using a monocular camera for obstacle detection. They first detect the horizon line and then search for a potential obstacle in the region below the horizon. A fundamental drawback of [15, 16] is that they approximate the edge of water by a horizon line and cannot handle situations in coastal waters, close to the shoreline or in marina. At that point, the edge of water does not correspond to the horizon anymore and can be no longer modeled as a straight line. Such cases call for more general segmentation approaches.

Many unsupervised segmentation approaches have been proposed in literature. Khan and Shah [18] use optical flow, color and spatial coordinates to construct features which are used in single Gaussians to segment a moving object in video. [19] have proposed a graph-theoretic clustering to perform segmentation of color images into visually-coherent regions. The assumption that the neighboring pixels likely belong to the same class is formally addressed in the context of Markov random fields (MRF) [20, 21]. [22] have extended the conditional random fields with dynamic models and perform the inference for object detection and labeling jointly in videos. The random field frameworks [23] have proven quite successful for addressing the semantic labeling tasks and recently [24] have shown that structural priors between classes further improve the labeling. The approaches like [22] use high-dimensional features composed

of color and texture at multiple scales and object-class specific detectors to segment the images and detect the objects of interest. In our scenarios, the possible types of dynamic obstacles are unknown and vary significantly in appearance. Thus object-class specific detectors are not suitable. Recently, Alpert et al. [25] have proposed an approach that starts from a pixel level and gradually constructs visually-homogenous regions by agglomerative clustering. They achieved impressive results on a segmentation dataset in which an object was occupying a significant portion of an image. Unfortunately, since their algorithm incrementally merges regions, it is too slow for online application even at moderate image sizes. An alternative to starting the segmentation from pixel level is to start from an oversegmented image such that pixels are grouped into superpixels [26]. Li et al. [27] have proposed a segmentation algorithm that uses multiple superpixel oversegmentations and merges their result by a bipartite graph partitioning to achieve state-of-the-art results on a standard segmentation dataset. However, no prior information is provided to favor certain types of segmentations in specific scenes.

1.2 Our approach and contributions

We pursue a solution for obstacle detection that is based on concepts of image segmentation with weak semantic priors on the expected scene composition. Figure 1 shows typical images captured from a USV. While the images significantly vary in appearance, we observe that each image can be split into three semantic regions roughly stacked one above the other, implying a structural relation between the regions. The bottom region represents the water, while the top region represents the sky. The middle component can represent either land, parked boats a haze above horizon or a mixture of these.

Our **main contribution** is a graphical model for structurally-constrained semantic segmentation with application to USV obstacle-map estimation. The generative model assumes a mixture model with three Gaussian components for the dominant three image regions and a uniform component for explaining the outliers, which may constitute an obstacle in the water. We propose a graphical model with weak priors on the mixture parameters and a MRF over the prior as well as posterior pixel-class distributions to favor smooth segmentations. We derive an EM algorithm for the proposed model and show that the resulting optimization achieves a fast convergence at a low computational cost, without resorting to a specialized hardware. A similar segmentation model was proposed in [28], but their model requires a manually set variable, does not apply priors and is not derived from a single density function.

We apply this model to obstacle image-map estimation in USVs. The proposed model acts directly on color image and does not require expensive extraction of texture-based features. Combined with efficient optimization, this results in faster than realtime segmentation and obstacle-map estimation. Our approach is outlined in Figure 1. The semantic model is fitted to the input image, after which each pixel is classified into one of the four classes. All the pixels that do not correspond to the water component are deemed to be a part of an obstacle.

Figure 1 shows a detection of a dynamic obstacle (buoy) and of a static obstacle (shoreline).

Our **second contribution** is a marine dataset for semantic segmentation and obstacle detection, and the performance evaluation methodology. To our knowledge this will be the largest annotated publicly available marine dataset of its kind up to date. The remainder of the paper is structured as follows. In Section 2 we derive our semantic generative model, in Section 3 we present the obstacle detection algorithm, in Section 4 we experimentally analyze the algorithm on an extensive dataset and draw conclusions in Section 5.

2 The semantic generative model

We consider the image as an array of measured values $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1:M}$, in which $\mathbf{y}_i \in \mathcal{R}^d$ is a d dimensional measurement, a feature vector, at the i-th pixel in an image with M pixels. As we detail in the subsequent sections, the feature vector is composed of pixel's color and image coordinates. The probability of the i-th pixel feature vector is modelled as a mixture model with four components – three Gaussians and a single uniform component:

$$p(\mathbf{y}_i|\Theta) = \sum_{k=1}^{3} \phi(\mathbf{y}_i|\mu_k, \Sigma_k)\pi_{ik} + \mathcal{U}(\mathbf{y}_i)\pi_{i4}, \tag{1}$$

where $\Theta = \{\mu_k, \Sigma_k\}_{k=1:3}$ are the means and covariances of the Gaussian kernels $\phi(\cdot|\mu, \Sigma)$ and $\mathcal{U}(\cdot)$ is a uniform distribution. The i-th pixel label x_i is an unobserved random variable governed by the class prior distribution $\pi_i = [\pi_{i1}, \ldots, \pi_{i4}]$ with $\pi_{i1} = p(x_i = i1)$. The three Gaussian components represent the three dominant semantic regions in the image, while the uniform component represents the outliers, i.e., pixels that do not likely correspond to any of the three structures. To encourage segmentations into three approximately vertically aligned semantic structures, we define a set of priors $\varphi_0 = \{\mu_{\mu_k}, \Sigma_{\mu_k}\}_{k=1:3}$ for the mean values of the Gaussians, i.e., $p(\Theta|\varphi_0) = \prod_{k=1}^{3} \phi(\mu_k|\mu_{\mu_k}, \Sigma_{\mu_k})$. To encourage smooth segmentations, the priors π_i as well as posteriors over the pixel class labels, are treated as random variables, which form a Markov random field. Imposing the MRF on the priors and posteriors rather than pixel labels allows effectively integrating out the labels, which leads to a well-behaved class of MRFs [28] that avoid image reconstruction during parameter learning. The resulting graphical model with priors is shown in Figure 2.

Let $\pi = \{\pi_i\}_{i=1:M}$ denote the set of priors for all pixels. Following [20] we approximate the joint distribution over the priors as $p(\pi) \approx \prod_i p(\pi_i|\pi_{N_i})$, and π_{N_i} is a mixture distribution over the priors of the i-th pixel's neighbors, i.e., $\pi_{N_i} = \sum_{j \in N_i, j \neq i} \lambda_{ij}\pi_j$, where λ_{ij} are fixed positive weights such that for each i-th pixel $\sum_j \lambda_{ij} = 1$. The potentials in the MRF are defined as $p(\pi_i|\pi_{N_i}) \propto \exp\left(-\frac{1}{2}E(\pi_i, \pi_{N_i})\right)$ with $E(\pi_i, \pi_{N_i}) = D(\pi_i \parallel \pi_{N_i}) + H(\pi_i)$. The term $D(\pi_i \parallel \pi_{N_i})$ is the Kullback-Leibler divergence which penalizes the differences between prior distributions over the neighboring pixels (π_i and π_{N_i}), while the term $H(\pi_i)$

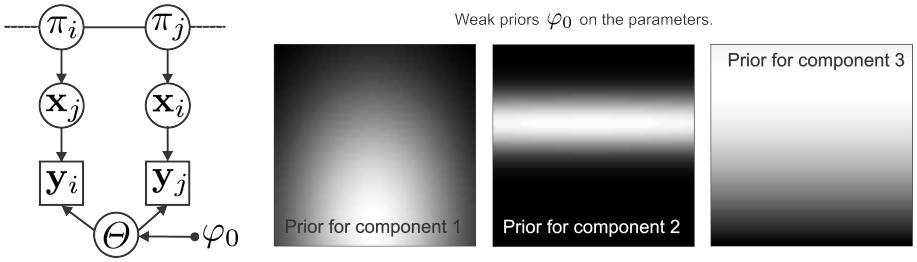

Fig. 2. The graphical model (left) with weak priors on three semantic components (right).

is the entropy and penalizes uninformative priors π_i. The joint distribution for the graphical model in Figure 2 can be written as

$$p(\mathbf{Y}, \Theta, \pi|\varphi_0) = \prod_{i=1}^{M} p(\mathbf{y}_i|\Theta, \varphi_0) p(\Theta|\varphi_0) p(\pi_i|\pi_{N_i}). \tag{2}$$

Diplaros et al. [28] argue that improved segmentations can be achieved by also considering an MRF directly on the pixel posterior distributions by treating the posteriors as random variables $\mathbf{P} = \{\mathbf{p}_i\}_{i=1:M}$, where the components of $\mathbf{p}_i$ are defined as $p_{ik} = p(x_i = k|\Theta, \mathbf{y}_i, \varphi_0)$, computed by Bayes rule from $p(y_i|x_i = k, \Theta)$ and $p(x_i = k)$. We can write the posterior over $\mathbf{P}$ as $p(\mathbf{P}|\mathbf{Y}, \Theta, \pi, \varphi_0) \propto \prod_{i=1}^{M} \exp(-\frac{1}{2} E(\mathbf{p}_i, \mathbf{p}_{N_i}))$, where $\mathbf{p}_{N_i}$ is a mixture defined in the same spirit as π_{N_i}. The joint distribution can now be written as

$$p(\mathbf{P}, \mathbf{Y}, \Theta, \pi|\varphi_0) \propto \exp[\sum_{i=1}^{M} \log p(\mathbf{y}_i, \Theta|\varphi_0) - \frac{1}{2}(E(\pi_i, \pi_{N_i}) + E(\mathbf{p}_i, \mathbf{p}_{N_i}))], \tag{3}$$

Due to coupling between π_i/π_{N_i} and $\mathbf{p}_i/\mathbf{p}_{N_i}$ the optimization of (3) is not straightforward. We therefore introduce auxiliary variables $\mathbf{q}_i$ and $\mathbf{s}_i$ and take the logarithm, which results in the following cost function

$$F = \sum_{i=1}^{M} [\log p(\mathbf{y}_i, \Theta|\varphi_0) - \frac{1}{2}(D(\mathbf{s}_i \| \pi_i \circ \pi_{N_i}) + D(\mathbf{q}_i \| \mathbf{p}_i \circ \mathbf{p}_{N_i}))], \tag{4}$$

where $\circ$ is the Hadamard (component-wise) product. Note that when $\mathbf{q}_i \equiv \mathbf{p}_i$ and $\mathbf{s}_i \equiv \pi_i$, (4) reduces to (3) (ignoring the constant terms). Maximization of F can now be achieved in an EM-like fashion. In the E-step we maximize F w.r.t. $\mathbf{q}_i$, $\mathbf{s}_i$, while the M-step maximizes over the parameters Θ and π. We can see from (4) that the F is maximized w.r.t $\mathbf{q}_i$ and $\mathbf{s}_i$ when the divergence terms vanish, therefore, $\mathbf{s}_i^{\text{opt}} = \xi_{s_i} \pi_i \circ \pi_{N_i}$, $\mathbf{q}_i^{\text{opt}} = \xi_{q_i} \mathbf{p}_i \circ \mathbf{p}_{N_i}$, where ξ_{s_i} and ξ_{q_i} are the normalization constants.

The M-step in not as straightforward, since direct optimization over Θ and π is intractable and we resort to maximizing its lower bound. We define $\hat{\mathbf{s}}_i =$

$(\mathbf{s}_i + \mathbf{s}_{N_i})$ and $\hat{\mathbf{q}}_i = (\mathbf{q}_i + \mathbf{q}_{N_i})$ and by Jensen's inequality lower-bound the divergence terms as

$$-D(\mathbf{s}_i \| \pi_i \circ \pi_{N_i}) \geq \hat{\mathbf{s}}_i^T \log \pi_i \; ; \; -D(\mathbf{q}_i \| \mathbf{p}_i \circ \mathbf{p}_{N_i}) \geq \hat{\mathbf{q}}_i^T \log \mathbf{p}_i, \tag{5}$$

where we have ignored the terms independent of π_i and $\mathbf{p}_i$. Substituting (5) into (4) and collecting the relevant terms yields the following lower bound on the cost function (4)

$$\hat{F} = \sum_{i=1}^{M} [\frac{1}{2}(\mathbf{q}_i + \mathbf{q}_{N_i})^T \log(\mathbf{p}_i p(\Theta|\varphi_0)) + \frac{1}{2}(\hat{\mathbf{s}}_i + \hat{\mathbf{q}}_i)^T \log \pi_i]. \tag{6}$$

Differentiating (6) w.r.t., π_i and applying a Lagrange multiplier with the constraint $\sum_k \pi_{ik} = 1$, we see that $\hat{F}$ is maximized at $\pi_i^{\text{opt}} = \frac{1}{4}(\hat{\mathbf{s}}_i + \hat{\mathbf{q}}_i)$. Differentiating (6) w.r.t. the means and covariances of Gaussians, we obtain

$$\mu_k^{\text{opt}} = \beta_k^{-1} [\Lambda_k (\sum_{i=1}^{M} \hat{q}_{ik} \mathbf{y}_i^T) \Sigma_k^{-1} - \mu_{\mu_k}^T \Sigma_{\mu_k}^{-1}]^T, \tag{7}$$

$$\Sigma_k^{\text{opt}} = \beta_k^{-1} \sum_{i=1}^{M} \hat{q}_{ik} (\mathbf{y}_i - \mu_k)(\mathbf{y}_i - \mu_k)^T, \tag{8}$$

where we have defined $\beta_k = \sum_{i=1}^{M} \hat{q}_{ik}$ and $\Lambda_k = (\Sigma_k^{-1} + \Sigma_{\mu_k}^{-1})^{-1}$. An appealing property of the model (4) is that its E-step can be efficiently implemented through convolutions and Hadamard products. Recall that the calculation of the i-th pixel's neighborhood prior distribution π_{N_i} entails a weighted combination of the neighboring pixel priors π_j. Let $\pi_{\cdot k}$ be the k-th component priors arranged in a matrix of image size. Then the neighborhood priors can be computed by the following convolution $\pi_{N \cdot k} = \pi_{\cdot k} * \lambda$, where λ is a discrete kernel with its central element set to zero and its elements summing to one. Let $\hat{\mathbf{s}}_{\cdot k}$, $\hat{\mathbf{q}}_{\cdot k}$ and $\mathbf{p}_{\cdot k}$ be the image-sized counterparts corresponding to sets of distributions $\{\hat{\mathbf{s}}_i\}_{i=1:M}$, $\{\hat{\mathbf{q}}_i\}_{i=1:M}$ and $\{\mathbf{p}_i\}_{i=1:M}$, respectively, and let λ_1 denote the kernel λ in which the central element is set to one. Then the calculation of the k-th component priors $\pi_{\cdot k}^{\text{opt}}$ for all pixels in the E-step can be written as

$$\begin{aligned} \hat{\mathbf{s}}_{\cdot k} &= (\xi_{s \cdot} \circ \pi_{\cdot k} \circ (\pi_{\cdot k} * \lambda)) * \lambda_1, \\ \hat{\mathbf{q}}_{\cdot k} &= (\xi_{q \cdot} \circ \mathbf{p}_{\cdot k} \circ (\mathbf{p}_{\cdot k} * \lambda)) * \lambda_1, \\ \pi_{\cdot k}^{\text{opt}} &= (\hat{\mathbf{s}}_{\cdot k} + \hat{\mathbf{q}}_{\cdot k})/4. \end{aligned} \tag{9}$$

The EM procedure for fitting our generative model to the input image is summarized in Algorithm 1.

3 Obstacle detection

We formulate the obstacle detection as a problem of estimating an image obstacle map, i.e., determining the pixels in the image that correspond to the sea while all

Algorithm 1 : The EM algorithm for the segmentation model.

Require:
Pixel features $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1:M}$, priors φ_0, initial values for Θ and π.

Ensure:
The estimated parameters π^{opt}, Θ^{opt} and the smoothed posterior $\{\hat{\mathbf{q}}_{\cdot k}\}_{k=1:4}$.

Procedure:
1: Calculate the pixel posteriors $\mathbf{p}_{\cdot k}$ using the current estimates of π and Θ for all k (1).
2: Calculate the new pixel priors $\pi_{\cdot k}^{\text{opt}}$ for all k using (9).
3: Calculate the new parameter values Θ using (7) and (8).
4: Iterate steps 1 to 3 until convergence.

the remaining pixels represent the potential obstacles. We therefore first fit our semantic model from Section 2 to the input image and estimate the smoothed a posteriori probability distribution $\hat{\mathbf{q}}_{ik}$ across the four semantic components for each pixel. An $i-$th pixel is classified as water if the corresponding posterior $\hat{\mathbf{q}}_{ik}$ reaches maximum for the water component among all four components. In our setting the component indexed by $k = 1$ corresponds to water region, which results in the labeled image B with the i-th pixel label b_i defined as

$$b_i = \begin{cases} 1 \ ; \arg\max_k \hat{\mathbf{q}}_{ik} = 1 \\ 0 \ ; \quad otherwise \end{cases} . \tag{10}$$

Retaining only the largest connected region in the image B results in the current obstacle image map $\hat{B}_t$. All blobs of non-water pixels within the connected water region are proclaimed as potential *obstacles in the water*. This is followed by a nonmaxima suppression stage which merges detections that are located in close proximities (e.g., due to object fragmentation) to reduce multiple detections of the same obstacle. The water edge is extracted as the longest connected outer edge of the connected region corresponding to the water. Note also that the Algorithm 1 requires initial values for the parameters Θ and π. We exploit the continuity of sequential images in the videostream by taking the parameter values of the converged model from the previous time-step for initialization of the EM algorithm in the current time-step. The obstacle detection is summarized in Algorithm 2 and visualized in Figure 1.

4 Experiments

4.1 Implementation details

In our application, the measurement at each pixel is encoded by a five-dimensional feature vector $\mathbf{y}_i = [i_x, i_y, i_h, i_s, i_v]$, where (i_x, i_y) are the i-th pixel coordinates and the (i_h, i_s, i_v) are the pixel's HSV color channels. We have also determined that we achieve sufficiently good obstacle detection by performing detection on a reduced-size image of 50×50 pixels and then rescale the results to the original image size. This drastically speeds up the algorithm to approximately 14ms per

Algorithm 2 : The obstacle image map estimation and obstacle detection.

Require:

Pixel features $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1:M}$, priors φ_0, estimated model from previous time-step Θ_{t-1} and $\hat{\mathbf{q}}_{t-1}$.

Ensure:

Obstacle image map $\hat{B}_t$, water edge $\mathbf{e}_t$, detected objects $\{\mathbf{o}_i\}_{i=1:N_{\text{obj}}}$, model parameters Θ_t and $\hat{\mathbf{q}}_t$.

Procedure:

1: Initialize the parameters of Θ_t and π_t by Θ_{t-1} and $\hat{\mathbf{q}}_{t-1}$.
2: Apply the Algorithm 1 and priors φ_0 to fit the model Θ_t and $\hat{\mathbf{q}}_t$ to the input data $\mathbf{Y}$.
3: Calculate the new obstacle image map $\hat{B}_t$ and for interpretation also the water edge $\mathbf{e}_t$ and the obstacles in water $\{\mathbf{o}_i\}_{i=1:N_{obj}}$.

frame in our experiments. The uniform distribution component in (1) is defined over the image pixels domain and returns equal probability for each pixel. In our rescaled image this means that $\mathcal{U}(\mathbf{y}_i) = \frac{1}{50^2}$ at each pixel. The only constraint on the convolution kernel λ (9) is that the central element is set to zero and all elements sum to one. We use a Gaussian kernel with central element set to zero and set the size of the kernel to 2% of image size, which results in a 3×3 pixels kernel. The spatial components in the feature vector play a dual role. On one hand they encode region texture through spatial correlation of colors. On the other hand they lend means to weakly constraining the Gaussian components such that they reflect the three dominant semantic image parts. This is achieved by the weak priors $p(\Theta|\varphi_0) = \prod_{k=1}^{3} \phi(\mu_k|\mu_{\mu_k}, \Sigma_{\mu_k})$ on the Gaussian means. The weak priors were estimated from a few typical images captured from the boat that highly varied in appearance and geometry and were not used for the testing phase. Figure 2 visualizes the spatial components of the weak priors. All parameters were kept constant in the experiments[1].

4.2 Marine obstacle detection dataset (Modd)

The marine obstacle detection dataset consists of 12 video sequences, providing in total 4454 fully annotated 640×480 frames. The video sequences have been recorded from different platforms, but from a vantage point that is consistent with the limitations of the small (under 2 meter) USV (see, e.g., Figure 1). The Axis 207W camera was placed approximately 0.7 m above the water surface, looking in front of the vehicle, with an approximately 55° field of view. Camera has been set up to automatically adjust to the variations in lighting conditions. Video sequences have been acquired on different times under different weather conditions. Each frame is annotated manually by a polygon denoting the edge of water and bounding boxes are placed on *large obstacles* (those that straddle

[1] For research purposes, we will provide the reference Matlab code of our approach, including the evaluation routines from the authors page.

the water edge) and *small obstacles* (those that are fully surrounded by water). See Figure 3 for illustration.

4.3 Performance evaluation

The performance evaluation methodology was designed to reflect the two distinct challenges that the USV faces: the water edge (shoreline or horizon) detection and obstacle detection. The former is measured as the root mean square error (RMSE) in water edge position (Edg), and the latter is measured via the efficiency of *small object* detection, expressed as precision ($Prec$), recall (Rec), F-score (F) and the average number of false positives per frame (aFP).

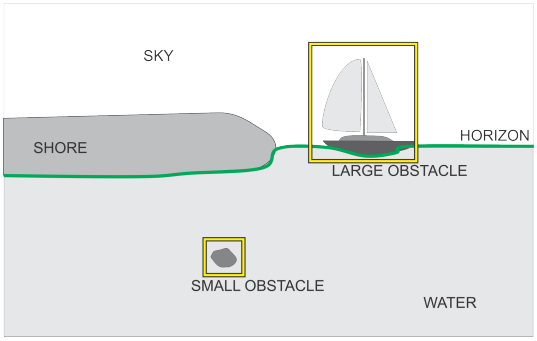

Fig. 3. Scene representation in Modd dataset.

The following protocol is used to evaluate RMSE in water edge estimation. Areas where *large obstacles* intersect the ground truth water edge are removed. Note that, given the scene representation (Figure 3), one cannot distinguish between large obstacles (e.g. large ships) and stationary elements of the shore (e.g. small piers). This way, a refined water edge was generated. For each pixel column in the full-sized image, a distance between water edge, as given by the ground truth and as determined by the algorithm is calculated. These values are averaged across all frames and videos and are shown in Table 1 as Edg.

The evaluation of object detection follows the recommendations from PASCAL VOC challenges [29], with small, application-specific modification: all small obstacles (provided as a ground truth or detected) that are closer to the annotated water line than 5% of image height, are discarded prior to evaluation on each frame. This was done to ensure fair competition in situations where a detection may oscillate between fully water-enclosed obstacle, and the "dent" in the shoreline. This is also consistent with the problem of obstacle avoidance – the USV is not concerned with avoiding small objects appearing right at the water edge. In counting false positives (FP), true positives (TP) and false negatives (FN), we followed the methodology of PASCAL VOC, with the minimum

overlap set to 0.3. FP, TP and FN are used to calculate precision ($Prec$), recall (Rec), F-score (F) and average false positives per frame (aFP).

4.4 Results

In the following we denote our semantic-segmentation-based obstacle image-map estimation algorithm as SSM. To evaluate how much each part of our model contributes to performance, we have also implemented two variants of our approach, which we denote by UGM and UGM_{col}. In contrast to SSM, the UGM and UGM_{col} do not use the MRF constraints and are in this respect only mixtures of a uniform pdf and three Gaussians with priors on their means. A further difference between UGM and UGM_{col} was that UGM_{col} ignored spatial information in visual features and relied only on color.

Note that SSM is conceptually similar to the Grab-cut algorithm [30], with two distinct differences. In contrast to the user-provided bounding box in [30], the SSM's weak supervision comes from the initialization of the parameters from the previous time-step and from the weak priors. The second distinction is that our approach does not explicitly calculate the segmentation mask to refine the mixture model. To further evaluate the MRF framework of our obstacle-map estimation algorithm, we have implemented a variant of our approach in which we apply a graph-cut [31] after each EM epoch to segment the image into a water/non-water mask. This mask is then used as in Grab-cut to refine the mixture model. We use exactly the same weakly-constrained mixture model as in SSM for fair comparison and denote this approach by GC. We have compared our approach also to the general segmentation approaches, namely the superpixel-based approach from Li et al. [27], SPX, and a graph-based segmentation algorithm from Felzenswalb and Huttenlocher [19], FZH. For fair comparison, all the algorithms were executed on the 50×50 images. We have experimented with the parameters of GC and FZH and have set them to optimal performance for our dataset. Since FZH was designed to run on larger images we have also performed the experiments for FZH on full-sized images – we denote this variant by FZH_{full}. All experiments were performed on a PC with 3.06 GHz Intel Xeon E5-1620 CPU in a single thread. The results are summarized in Table 1.

A Matlab implementation of SSM performed at rate higher 70 frames per second. Most of the processing was spent on fitting our semantic model and obstacle-map estimation (10ms), while 4ms was spent on the obstacle detection. For fair comparison of segmentation algorithms, we report in the Table 1 only the times required for the obstacle-map estimation. Although note that the obstacle detection part did require more processing time for the methods that delivered poor segmentation masks with more false positives. On average, our EM algorithm in SSM converged in three iterations. Note that the graph cut routine in GC SPX and the FZH were implemented in C and interfaced to Matlab, while all the other variants were entirely implemented in Matlab. Therefore, the computational time results for segmentations are not directly comparable among the methods, but still offer a level of insight.

Table 1. Performance evaluation on Modd. The table shows edge of water estimation error, precision, recall, F measure, average false positives and segmentation time, denoted by Edg, Prec, Rec, F, aFP, Time, respectively. The brackets show standard deviation where available.

	Edg[pix]	Prec	Rec	F	aFP	Time[ms]
SSM	10.8(8.9)	0.794	0.771	0.771	0.062	10(2)
GC	28.0(23.4)	0.555	0.736	0.606	0.348	15(3)
UGM	30.2(23.6)	0.524	0.738	0.575	0.525	11(2)
UGM_{col}	31.5(22.5)	0.118	0.490	0.177	2.692	11(3)
FZH	86.0(62.0)	0.728	0.525	0.554	0.043	16(1)
FZH_{full}	36.2(40.9)	0.440	0.802	0.529	0.621	200(3)
SPX	63.6(35.3)	0.007	0.001	0.001	0.079	55(1)

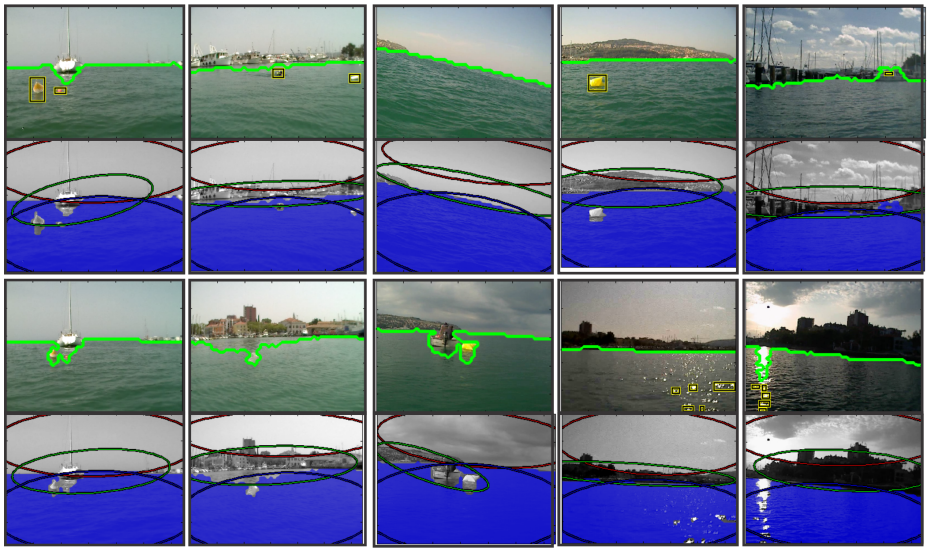

Fig. 4. Examples of water segmentation and obstacle detection. The detected edge-of-water is shown in green, while obstacles are shown as yellow rectangles. For each image we also show the spatial part of the three semantic components as three Gaussian ellipses and the portion of the image segmented as water in blue. Failure cases are shown in the bottom row with miss- and false detections.

From the results in Table 1 we can observe that the SSM outperformed all competing approaches by all measures. When FZH_{full} was run on full-sized images its recall improved compared to 50×50 image size version, but precision of object detection decreased, the false positive rate increased and the processing speed decreased by a factor of 10. Compared to GC, our approach delivered superior performance by all measures at comparable speed. This speaks of advantage of the continous optimization in the MRF used in our model compared to the standard MRF on pixel labels that requires binarization by cuts. By far the worst performance was for the SPX, the reason being that the resulting segmentations were too general for the problem at hand.

The improved performance of SSM can be attributed exclusively to our carefully designed graphical model. This is evident from the results of UGM in which we have ignored the MRF constraints. We observe a significant drop in performance, especially precision. The performance further drops with UGM_{col}, which implies that spatial components in the feature vectors bear important information for proper segmentation. Figure 4 shows examples of segmentation maps from our approach, the spatial part of the Gaussian mixture and the detected objects in water. The appearance of water varies significantly between the various scenes, and the same is true for the other two semantic components. The images also vary in the scene composition in that the vertical position as well as the attitude of the water edge vary significantly. Nevertheless, the model is able to adapt well to these compositions and successfully decomposes the scene into obstacles and fairly well delineates the water edge. The bottom row shows failure cases. The first three images show failure when the object in water is detected as part of the above-water region. Note that in these cases the USV will still successfully avoid collision, but such detection represents a false negative in our performance evaluation. The rightmost two images show the performance when the boat is facing direct sunlight that causes significant glitters on the water surface. Even in these harsh conditions the model is able to interpret the scene well enough with few false obstacle detections.

5 Discussion and conclusion

A graphical model for semantic segmentation of marine scenes was presented and applied to USV obstacle-map estimation. The model exploits the fact that scenes a USV encounters may be decomposed into three dominant visually- and semantically-distinctive components, one of which is the water. The appearance is modelled by a mixture of Gaussians and accounts for the outliers by a uniform component. The geometric structure is enforced by placing weak priors over the component means. A MRF model is applied on prior and posterior pixel-label distribution to account for the interactions across neighboring pixels. An EM algorithm is derived for fitting the model to image, which affords fast convergence and efficient implementation. The proposed model directly applies straight-forward features, i.e., color channels and pixel positions and avoids potentially slow extraction of more complex features. Nevertheless, the model is

general enough to be directly applied without modifications to any other features. Results show excellent performance compared to related segmentation approaches and exhibits improved performance in terms of segmentation accuracy as well as speed.

Note that [32] have proposed an approach for inference in image segmentation that segments urban area images into three-strip segmentations by a dynamic program. In contrast to our approach, [32] only address the labeling part of the segmentation and require precomputed per-pixel label confidences. The resulting segmentation contains a homogenous bottom region, which prevents detection of obstacles without further re-processing the features of the bottom pixels. Our approach jointly learns the component appearance, estimates the per-pixel class probabilities, and estimates the segmentation within a single online framework, by optimizing a well-defined graphical model. Some related maritime segmentation approaches [15, 16, 7] rely on good horizon estimation to approximate the water edge, which makes them inapplicable to coastal regions. Note that in coastal regions, the water edge does not correspond to horizon and due to variety of shore line and piers takes shapes far from a straight line. Our graphical model does not make such a strict assumption which makes it applicable to off-shore as well as coastal regions. Nevertheless, the graphical model is still general enough to enable direct incorporation of externally measured horizon line along with its uncertainty if available.

As our second contribution, we have presented a new real-life marine segmentation dataset. This will be the largest publicly-available dataset of its kind to date. The experimental results show that the proposed algorithm performs favorably compared to the related solutions. While the algorithm provides high detection rates at low false positives it does so with a low processing time (our current C++ implementation of SSM runs close to 200fps). Fast performance is of crucial importance for real-life implementations on USVs, as it allows the use in onboard embedded controllers and low-cost embedded, low-resolution cameras. In future work we will explore possibilities of porting our algorithm to such an embedded sensor. Since our optimization can be highly parallelized, we will explore this avenue in GPUs, which are becoming increasingly present in many modern embedded devices. Another avenue of further research will be analysis of additional low-level features for computation of better segmentation, addition of other modalities and extension to fast stereo systems, which may be feasible due to considerable computational speed of the proposed algorithm.

Acknowledgment

This work was supported in part by the Slovenian research agency programs P2-0214, P2-0094, and projects J2-4284, J2-3607, J2-2221. We also thank HarphaSea d.o.o. for their hardware used to capture the dataset.

References

1. Heidarsson, H., Sukhatme, G.: Obstacle detection from overhead imagery using self-supervised learning for autonomous surface vehicles. In: Int. Conf. Intell. Robots and Systems. (2011) 3160–3165
2. Rasmussen, C., Lu, Y., Kocamaz, M.K.: Trail following with omnidirectional vision. In: Int. Conf. Intell. Robots and Systems. (2010) 829 – 836
3. Montemerlo, M., Thrun, S., Dahlkamp, H., Stavens, D.: Winning the darpa grand challenge with an ai robot. In: AAAI Nat. Conf. Art. Intelligence. (2006) 17–20
4. Dahlkamp, H., Kaehler, A., Stavens, D., Thrun, S., Bradski, G.: Self-supervised monocular road detection in desert terrain. In: RSS, Philadelphia, USA (2006)
5. Ettinger, S.M., Nechyba, M.C., Ifju, P.G., Waszak, M.: Vision-guided flight stability and control for micro air vehicles. Advanced Robotics **17** (2003) 617–640
6. Lu, Y., Rasmussen, C.: Simplified markov random fields for efficient semantic labeling of 3D point clouds. In: IROS. (2012)
7. Scherer, S., Rehder, J., Achar, S., Cover, H., Chambers, A., Nuske, S., Singh, S.: River mapping from a flying robot: state estimation, river detection, and obstacle mapping. Auton. Robots **33** (2012) 189–214
8. Onunka, C., Bright, G.: Autonomous marine craft navigation: On the study of radar obstacle detection. In: ICARCV. (2010) 567–572
9. Heidarsson, H., Sukhatme, G.: Obstacle detection and avoidance for an autonomous surface vehicle using a profiling sonar. In: ICRA. (2011) 731–736
10. Rankin, A., Matthies, L.: Daytime water detection based on color variation. In: Int. Conf. Intell. Robots and Systems. (2010) 215–221
11. Elkins, L., Sellers, D., Reynolds, W.M.: The autonomous maritime navigation (amn) project: Field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles. Journal of Field Robotics **27** (2010) 790818
12. Hong, T.H., Rasmussen, C., Chang, T., Shneier, M.: Fusing ladar and color image information for mobile robot feature detection and tracking. In: IAS. (2002)
13. Santana, P., Mendica, R., Barata, J.: Water detection with segmentation guided dynamic texture recognition. In: IEEE Robotics and Biomimetics (ROBIO). (2012)
14. Socek, D., Culibrk, D., Marques, O., Kalva, H., Furht, B.: A hybrid color-based foreground object detection method for automated marine surveillance. In: Advanced Concepts for Intelligent Vision Systems, Springer (2005) 340–347
15. Fefilatyev, S., Goldgof, D.: Detection and tracking of marine vehicles in video. In: Proc. Int. Conf. Pattern Recognition. (2008) 1–4
16. Wang, H., Wei, Z., Wang, S., Ow, C., Ho, K., Feng, B.: A vision-based obstacle detection system for unmanned surface vehicle. In: Int. Conf. Robotics, Aut. Mechatronics. (2011) 364–369
17. Huntsberger, T., Aghazarian, H., Howard, A., Trotz, D.C.: Stereo visionbased navigation for autonomous surface vessels. JFR **28** (2011) 3–18
18. Khan, S., Shah, M.: Object based segmentation of video using color, motion and spatial information. In: Comp. Vis. Patt. Recognition. Volume 2. (2001) 746–751
19. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. Int. J. Comput. Vision **59** (2004) 167–181
20. Besag, J.: On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society **48** (1986) 259–302
21. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient nd image segmentation. International Journal of Computer Vision **70** (2006) 109–131

22. Wojek, C., Schiele, B.: A dynamic conditional random field model for joint labeling of object and scene classes. In: ECCV. (2008) 733 – 747
23. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. Int. Conf. Mach. Learning. (2001) 282 – 289
24. Kontschieder, P., Bulo, S., Bischof, H., Pelillo, M.: Structured class-labels in random forests for semantic image labelling. In: ICCV. (2011) 2190–2197
25. Alpert., S.Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: CVPR. (2012) 1–8
26. Ren, X., Malik, J.: Learning a classification model for segmentation. In: ICCV. (2003) 10 – 17
27. Li, Z., , Wu, X.M., Chang, S.F.: Segmentation using superpixels: A bipartite graph partitioning approach. In: CVPR. (2012)
28. Diplaros, A., Vlassis, N., Gevers, T.: A spatially constrained generative model and an em algorithm for image segmentation. IEEETNN **18** (2007) 798 – 808
29. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV **88** (2010) 303–338
30. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. In: SIGGRAPH. Volume 23. (2004) 309–314
31. Bagon, S.: Matlab wrapper for graph cut (2006)
32. Felzenszwalb, P.F., Veksler, O.: Tiered scene labeling with dynamic programming. In: CVPR. (2010)