https://doi.org/10.2352/ISSN.2470-1173.2017.13.IPAS-205

Water Region Extraction in Thermal and RGB Sequences Using Spatiotemporally-Oriented Energy Features

Amir Ghahremani, Egor Bondarev, Peter H.N. de With; Department of Electrical Engineering, Eindhoven University of Technology; Eindhoven, The Netherlands

Abstract

Although the concept of Regions Of Interest (ROI) is known in video analysis, the ROI extraction problem has been hardly addressed in maritime surveillance, particularly for vessel detection and tracking. A video captured by a maritime surveillance camera may contain irrelevant regions, such as shorelines, bridges and piers. As a result, non-relevant moving objects (e.g. cars moving along the shorelines) can be misleadingly detected by the vessel or ship surveillance system. This paper proposes a robust water region extraction method based on spatiotemporally-oriented energy features in combination with a mean shift clustering algorithm. The method targets not only the conventional RGB surveillance data, but also data from thermal cameras. Experimental results reveal that the pixel-wise water segmentation recall is 95.23% on average for the RGB images and 94.29% on average for the thermal images, even in the presence of islands or other complex shoreline shapes. The measured average precisions are 93.88% and 95.41% for the RGB and thermal datasets, respectively.

Introduction

Maritime surveillance is an important research topic, considering that it is vital to keep the maritime environment safe against dangers like terroristic attacks and illegal activities such as out-of-region-fishery, human and drugs traffic. In a general maritime surveillance system, besides radar systems in large harbors, visual cameras are deployed along the shorelines which capture the videos of the maritime environment. As the main task for such a system, vessels moving in the maritime region should be detected and their behavior analyzed. However, while aiming at finding maritime vessels, the surveillance cameras also capture multiple non-relevant regions in the scene, containing lots of moving objects, such as humans, trees, cars and clouds. As a result, sub-optimal ship detectors may extract a lot of non-relevant objects as maritime vessels. Consequently, the surveillance system is supposed to segment the maritime region and then only analyze the objects located in the extracted water region.

Despite the importance of ROI detection for maritime surveillance, at present, there were no robust methods proposed for maritime region detection. The main strategy explored in literature is detection of the horizon line to make the ROI smaller [1–7]. Consequently, such methods can primarily deal with simple scenarios, where there are only sea and sky regions in the captured scene. Besides this, there are several methods that do not attempt to find the ROI at all [8–14]. Such methods try to model the background (which is mostly the water itself) and then extract the target object directly by background subtraction. However, by evaluating the empirical results presented by the mentioned methods, we could not find work that illustrates object detection output on frames with a dynamic background and moreover, the test scenes contain generally only vessels as moving objects. Consequently, when these methods are applied to a scene containing a dynamic shoreline and city harbor regions with a lot of cars, busses, pedestrians, etc., the detection output would include all the moving objects at the shoreline as well. To address this issue, in [15], a water-region detection method is proposed. This work first performs a graph-based segmentation and then detects the water regions, using an off-line trained SVM. The work in [16,17] train a supervised classifier to perform pixel-based water detection. These methods make use of color, texture and spatiotemporal statistics of images to form the feature vector. In [18], after a pre-processing step focusing on increasing the invariance against water reflections and colors, spatiotemporal descriptors are used to locally classify the presence of water. This work generates a water detection mask through spatiotemporal Markov Random Field regularization of the local classifications. However, all these methods are evaluated just on simple cases. In our research for the industry-oriented European APPS project, we consider even more complex scenarios. For example, maritime regions with circular shorelines that include islands containing lots of non-relevant moving objects.

Spatiotemporally-oriented energy features, extracted through a 3D filtering approach [19–21], provide a rich representation of pixels. Such a representation describes both the static and dynamic aspects of the spatiotemporal behavior of pixels. These features have been already used in many applications. In [22, 23], authors use these features for object tracking. The method described in [24] benefits from energy features for dynamic texture recognition tasks. In [25], these features are used along with a mean shift clustering algorithm to group coherent regions. However, energy features are not adapted for maritime surveillance yet. In our case, we aim at solving the ROI detection problem for maritime surveillance, and propose a method that deploys the spatiotemporally-oriented energy representation of pixels for robust water detection, for both visual camera and thermal camera signals.

This paper is organized as follows. Section 2 explains the proposed method. Section 3 presents the experimental results validation. Section 4 concludes the paper.

Water Segmentation Pipeline

This section provides a brief overview on the water-region extraction pipeline and explain its individual algorithms. The method consists of the following four steps. First, a histogram of energy features for each pixel is extracted. Second, the extracted feature space is smoothened using the mean shift algorithm [25].

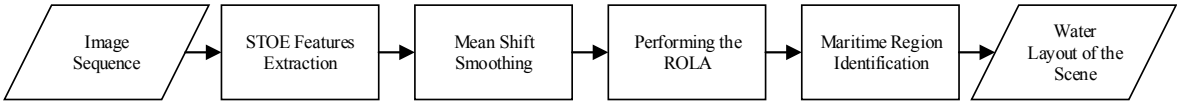


Figure 1: Architecture pipeline of the water segmentation method.

Third, based on the smoothed feature vectors, pixels are grouped into coherent regions of structures. Finally, our maritime region identification criteria are applied as discussed above. In this step, the energy histograms of pixels are accumulated to generate an energy histogram for each cluster. Then, by jointly exploring the bins of those histograms, clusters are detected that can be characterized as a maritime region. The detected regions that do not satisfy a minimum-size criterion, are omitted. Fig. 1 illustrates the proposed pipeline.

A. Spatiotemporally-Oriented energy Features

Objects can move in arbitrary directions in a scene, providing specific spatiotemporal data structures for image regions during consecutive frames. Therefore, these regions may have motion in any direction. Consequently, static 2D structures and/or dynamic structures (e.g. flicker [19]) can appear anywhere and be distributed over the region characteristics. Such spatiotemporal structures can potentially appear in all pixels located inside the region. Consequently, a discrete histogram can be constructed for each pixel, which represents its spatiotemporal behavior. To this end, the amount of the aforementioned 2D or 3D structures contained by a pixel should be measured and collected together.

The above-explained concept is discussed in [19–26], where the mentioned responses are described as spatiotemporally-oriented energy features. The measuring procedure would be performed through a set of 3D filter banks.

This paper deploys broadly tuned, steerable, separable 3D Gaussian second-derivative filters (G_2) and their Hilbert transforms (H_2) [20,21], where responses are point-wise rectified and summed. The local energy $E(\mathbf{x})$ for a pixel can be specified by the following equation:

$$E(\mathbf{x};\,\theta) = [G_2(\theta) * I(\mathbf{x})]^2 + [H_2(\theta) * I(\mathbf{x})]^2 \,, \tag{1}$$

where $\mathbf{x} = (x,y,t)$ indicates the pixel coordinates, the symbol * denotes the convolution operator and $I(.)$ is the current video frame from the image sequence. The parameter θ denotes the orientation of the filters used for energy feature extraction, i.e. the 3D direction of the filter axes of symmetry. Unfortunately, the energy measurements obtained by Eq. (1) are dependent on the image contrast level. To solve this, energies need to be normalized [23] with the following equation:

$$\hat{E}(\mathbf{x};\,\theta) = \frac{E(\mathbf{x};\,\theta)}{\sum_{\hat{\theta}} E(\mathbf{x};\,\hat{\theta}) \,+\, \varepsilon} \,, \tag{2}$$

where ε is a constant to avoid instabilities when the energy values are small and the summations in the denominator are performed over all considered orientations.

The discussed SpatioTemporally-Oriented Energy (abbreviated further as STOE) features describe pixels according to their static and dynamic characteristics. However, the STOE features of the pixels without special structure (e.g. pixels of a blue sky) would not contain any specific information. These pixels can be characterized by a lack-of-structure feature [26], calculated according to Eq. (3), given by

$$\hat{E}(\mathbf{x};\,\varepsilon) = \frac{\varepsilon}{\sum_{\hat{\theta}} E(\mathbf{x};\,\hat{\theta}) \,+\, \varepsilon} \,. \tag{3}$$

A key aspect of the STOE features defined in Equations (1)–(3), is the use of point-wise linear (i.e. here separable convolution and addition) and point-wise non-linear operations (i.e. here squaring and division), which intrinsically leads to a computationally efficient realization [23]. Meanwhile, due to the band-pass nature of used filters, extracted energies will be invariant to the additive image intensity variations [23].

All components of the scene have their own spatiotemporal characteristics. As mentioned above, STOE features are rich spatiotemporal behavioral descriptors for pixels and regions. Therefore, the proposed method performs segmentation of maritime/water regions using the STOE features. In general, it is not possible to detect a specific region by using just one STOE feature, since in most cases there are multiple types of components sharing the same specific characteristic(s) in the scene (e.g. a high-energy feature value in flicker). However, it seems to be possible to identify a specific region by jointly exploring several STOE features simultaneously, which is the main concept for the maritime region segmentation in this paper. The method proposes to represent each pixel by 8 energy features: 2 static horizontal and vertical orientations, 5 dynamic orientations (flicker, rightward, leftward, upward, and downward motion) and 1 so-called lack-of-structure feature.

Although these features contain useful information about spatiotemporal behavior of pixels, they lack localization information. This is because energy features are outputs of a spatiotemporal filtering stage. Consequently, they rely on neighboring pixels gray-level values. As a result, two pixels belonging to the same object (although containing the same spatiotemporal behavior) may have different energy representation. Therefore, the sharp discontinuities in energy representation of neighboring pixels should be smoothened, to attenuate the noise and enhance the spatiotemporal coherency between pixels of a specific region which contain similar spatiotemporal behavior. To implement this, we deploy a mean shift clustering method.

B. Mean Shift Algorithm

Feature space analysis is a useful approach to investigate characteristics of a scene. This paper uses STOE features to represent static and dynamic pixel characteristics. There are several methods to perform the feature space analysis. One well-known method is the mean shift algorithm, which regards the feature space as an empirical distribution [27]. This subsection provides a brief overview on mean shift, considering that the STOE features are applied to the mean shift framework to cluster the scene into coherent regions of structure. This method tries to smooth the feature space by associating feature points with a mode during an iterative procedure. The algorithm would stop when all feature

points associated with a particular mode, become close enough to each other in the feature space and share a common value. Mean shift is a nonparametric clustering method, thus it does not require any prior knowledge about the amount and shape of clusters.

Hence, mean shift replaces the feature vector of each pixel with the mean of feature vectors of similar pixels located in the neighborhood of that pixel. From this point of view, the mean shift algorithm requires three inputs: (a) radius *msR*, specifying candidate pixels for calculation of mean value for the current pixel, (b) similarity threshold *msSimTr*, determining which pixels located inside the radius *msR* are allowed to take part in calculation of mean value for the current pixel, and (c) convergence threshold *msCngTr*, specifying when (i.e. after how many iterations) the convergence occurs for each pixel and the mean shift procedure should stop.

After performing the mean shift smoothing algorithm, pixels with similar feature vectors are grouped together in order to form clusters. The proposed grouping method is presented in the following section.

C. Labeling Algorithm

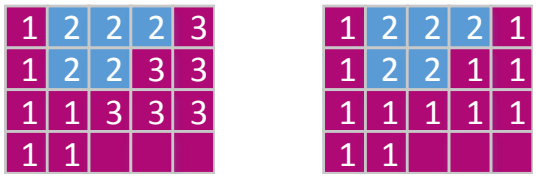
After the mean shift algorithm, pixels sharing similar feature vectors are clustered into coherent regions of structures. In this paper, we propose a Raster Order based Labeling Algorithm (ROLA) assigning the same cluster label to similar pixels that are just one pixel (horizontally and/or vertically) distant from each other.

ROLA consists of two main phases: label assignment and label correction. The method starts from the pixel located at the top-left corner of an image and continues towards the right until the first row is completely labeled. Then it repeats for succeeding rows.

The method assigns the label zero to the first pixel and increases the label counter by one. The method compares the current pixel with its four adjacent pixels, located at the right, down-left, down, and down-right sides (if existing). The comparison method can be selected based on the type of feature vectors. Here, we represent pixels with spatiotemporal energy histograms and therefore, measure their similarity using the Bhattacharyya coefficient [28]. However, when describing pixels based on other feature vectors (e.g. RGB values), the similarity can be measured with other metrics like Euclidean distance.

After measuring similarities, the method assigns label numbers to the adjacent pixels. If the similarity of neighboring pixels is higher than a threshold *grpSimTr*, then the neighbor will preempt the label from the current pixel. Otherwise, it will be left without any label. Now, the labeling pointer moves to the next pixel. If the next pixel has no label assigned, it would take the new label number stored in the label counter and the counter would be incremented. However, if the new pixel already has a label, the ROLA would proceed to the comparison iteration for that pixel and try to assign the label to its neighbors.

During the comparison step, the method may face a situation in which an adjacent pixel is already labeled with a different label than the current pixel label, but has proper similarity to the current pixel higher than the threshold *grpSimTr*. This is a correction case which occurs when previously labelled non-adjacent pixels inherently belong to the same cluster, but have been labeled to different clusters, due to the horizontally-directed labeling procedure. However, at the current row, these two clusters with different labels become adjacent to each other. Fig. 2 illustrates the correction case. In Fig. 2a, pixels that have the same color are similar pixels (in this paper, according to their energy histograms) and should finally be clustered together, since they have just one pixel distance to each other. Pixels labeled as unity and those that have the label 3 are inherently belonging to the same cluster. However, they have different labels and should be incorporated to form a unified cluster. The method detects this error when trying to propagate the label of the second unity in the third row and enters the correction phase to solve the problem. The corrected region is shown in Fig. 2b.



(a) (b)

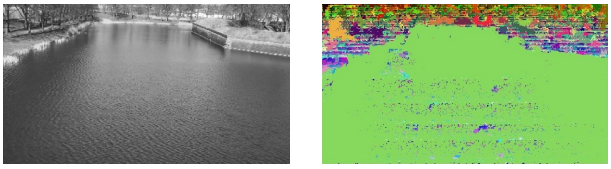
Figure 2: Example of the correction phase of the ROLA grouping method.

In the correction phase, the ROLA tries to incorporate the incorrectly assigned clusters. The algorithm takes both the labels of the current pixel and that adjacent pixel, and then assigns the smaller label to all pixels of the cluster with the higher label. Afterwards, values of other labels higher than the eliminated label are reduced by one. The algorithm finalizes the correction phase by reducing the label counter by one.

ROLA is specified in a pseudo code given below. The presented algorithm groups the neighboring similar pixels. The *SimilarityFunction* used in the code is a function which measures similarity between pixels. In this paper, we use the Bhattacharyya coefficient, considering that we represent pixels with energy features. However, based on the application, other metrics could be deployed as well. Fig. 3 illustrates the performance of the grouping method.

D. Maritime Region Identification

In maritime surveillance, all mobile objects on the water should be detected and tracked. However, a scene captured by a surveillance camera normally includes some parts of the background (e.g. shorelines, ports, buildings, vegetation) as well. Consequently, the detection system may occasionally consider a non-relevant object moving on the background as an object of interest. Therefore, it is important to first detect and extract the maritime region of the scene. Then, detection and tracking can be applied to that region and the moving objects located within the extracted region can be further examined.



(a) (b)

Figure 3: Grouping method performance.

Algorithm 1 ROLA algorithm in pseudo-code.

Initialize $LabelCounter = 0$, and $Label(1:M, 1:N) = -1$, where M and N are the number of rows and columns of image, respectively.
for $((row = 1:M)$ and $(pixel = 1:N))$ **do**
  if $(Label(row,\ pixel) == -1)$ **then**
    $LabelsTable(row, pixel) = LabelCounter$; $LabelCounter = LabelCounter + 1$;
  else
    for $(i = 1:4)$ **do**
      R, and P = coordinates of Right, Down-Left, Down, and Down-Right adjacent pixels; $[SimVal] = SimilarityFunction(FeaturePoint(row, pixel), FeaturePoint(R,P))$;
      if $(SimVal > grpSimTr)$ **then**
        if $(Label(R,P) == -1)$ **then**
          $Label(R,P) = Label(row, pixel)$;
        else
          $[Label, LabelCounter] = CorrectionFunction(Label, row, pixel, R, P, LabelCounter)$;
function $[Label, LabelCounter] =$ **CorrectionFunction**$(Label, row, pixel, R, P, LabelCounter)$
for $(x = 1:row+1)$ and $(y = 1:N)$ **do**
  if $Label(x,y) ==$ **max**$(Label(row, pixel), Label(R,P))$ **then**
    $Label(x,y) =$ **min**$(Label(row, pixel), Label(R,P))$;
  else if $(Label(x,y) >$ **max**$(Label(row, pixel), Label(R,P)))$ **then**
    $Label(x,y) = Label(x,y) - 1$;
$LabelCounter = LabelCounter - 1$;
end **function**

In this paper, we propose a method to extract the water part of the scene. To this end, we choose to represent the pixels of the image by their energy features, since these features describe both static and dynamic attributes of pixels. This property plays an important role in our method, considering that the maritime region differentiates itself from its surroundings with intrinsic dynamic behavior. Continuously moving waves feed this intrinsic dynamic behavior, which can be rarely found in background parts. Prior to energy extraction, we perform temporal subsampling of the video sequences, considering that this makes maritime region behavior even more dynamic.

After extracting the energy representation of pixels according to Section 2.A, we use the mean shift filtering frame work to smoothen the feature vectors of pixels. Then, the filtered image is clustered into coherent regions of structures by applying the grouping algorithm explained in previous section.

The core function to this step accumulates the energy histograms of all pixels located inside each cluster in order to make an energy histogram for that cluster. However, for RGB sequences, we horizontally divide the scene into 3 equal zones and experimentally multiply the flicker bins of two upper zones by 2, considering that those zones appertain to far places where the flicker bin of pixels do not have comparative amounts to the near pixels. For thermal images, we apply this to pixels which are located between 0.4 and 0.75 of the image height. The maritime clusters are found according to their energy representations. The Eq. 1 specifies the criteria for maritime region detection. If the sum of two static features per number of cluster pixels is between the thresholds *StaticLtr* and *StaticHtr* and the amount of the flicker bin per number of cluster pixels is higher than the threshold *FlickerTr*, the cluster is considered to belong to a maritime region. However, if both static horizontal and vertical bins of a cluster are exceeding the threshold *StaticHrVrTr*, the cluster would be considered as a background, since static structures inside the water part can be created by just horizontally or vertically structured waves. Additionally, the lack-of-structure bin has to be larger than the threshold *UnstrctrTr*. As the output of this step, the method makes a water map for the frame, by assigning a unity value to pixels that are distinguished as "maritime and assign zero value to pixels that are distinguished as "background.

Statement 1:
$\{StaticLtr < (EH(x,y,1) + EH(x,y,2)) < StaticHtr\}$ *and*
$\{(EH(x,y,t,3) > FlickerTr)\}$ *and* $\{(EH(x,y,t,8) > UnstrctrTr)\}$

Statement 2:
$\{(EH(x,y,t,1) > StaticHrVrTr)\}$ *and*
$\{(EH(x,y,t,2) > StaticHrVrTr)\}$

$$WaterMap(x,y) = \begin{cases} 1, & \text{(statement 1 == true)} \\ & \textit{and} \text{ (statement 2 == false);} \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

The method may result in a few false positive detections, when background is classified as a water area. Likewise, false negatives may also occur, when some regions inside maritime region are detected as non-maritime. However, such regions are small and are generated because of temporary changes in illumination, appearance of objects, waves, moving ships, etc. To solve this problem, we propose two techniques.

First, we invert all regions which contain less than D pixels. With this technique, the background regions falsely marked as positive, change to negative, while maritime regions marked as negative, change to positive. As a result, the technique merges all small noise regions with their surroundings. The second technique mostly solves the problem caused by big ships which are present in the water. Evidently, according to Eq. (1), ships are never detected as maritime region, but instead generate large holes in the detected water region. However, ships are mobile and rarely stay stationary at one place for a long time. Therefore, we propose to repeat the whole algorithm for T times during the captured video and calculate the mean value of created water maps. After this temporal averaging technique, if the value of a final water-map pixel is higher than the threshold *MapMeanTr*, we conclude that the pixel belongs to the maritime region, and mark it as background pixel otherwise.

Empirical Validation

In this section, we validate the extraction method on 22 RGB and 17 thermal image sequences. The constant ε in Equations (2)–(3) is set to 1% of the maximum energy values among all the energies. Oriented energies are extracted using basic filters with the kernel size of $5 \times 5 \times 5$ pixels (i.e. height $\times$ width $\times$ depth). The sequences are captured with the frame rate of 25 fps and with the resolution of 1924×1080 pixels. We perform frame sub-sampling by a factor of 25 (after this temporal subsampling, the method operates on 1-fps sequences) to make the dynamic water behavior more discriminative and spatially downsample the sequences by a factor of 4 to decrease the computational burden.

In the mean shift filtering stage, a window of $11 \times 11 \times 3$ pixel size (i.e. height $\times$ width $\times$ depth) specifies the neigh-

borhood of each pixel. Other parameters of the mean shift algorithm for RGB videos are set to: *msSimTr*=0.9 and *msCngTr*=0.99. The *grpSimTr* threshold used in the ROLA grouping is set to 0.9995. Other method parameters specified in the previous section are set as: *StaticLtr*=0.01, *StaticHtr*=0.3, *FlickerTr*=0.03, *UnstrctrTr*=0, *StaticHrVrTr*=0.1, *D*= 400, *MapMeanTr*= 0.6. For thermal videos, we reuse the RGB parameter values except for: *msSimTr*=0.92 and *msCngTr*=0.99, *grpSimTr*=0.9997, *StaticLtr*=0, *StaticHtr*=0.1, *FlickerTr*=0.01, and *UnstrctrTr*=0.3.

Fig. 4 and Fig. 5 illustrate the results of the method on 6 RGB and 6 thermal sequences, respectively. In each row, the original image and the extracted water region(s) are depicted. Although maritime surveillance cameras are commonly positioned to capture mostly the water ways, we have validated the method on non-standard challenging sequences containing islands and several types of shorelines (city, vegetation), to illustrate the robustness of the method in water region extraction.

For vessel detection applications, a pixel-level accuracy of the water-region extraction is not required and an approximate layout of the water regions is sufficient. Since vessels occupy a significant amount of pixels in a scene, small parts of water ways, missing in the extracted water map, would not hinder the ship detection. Therefore, although the extracted water regions do not exactly fit to the ground truth maps, they provide sufficient information to differentiate vessels from irrelevant moving objects.

Tables 1 and 2 present quantitative analysis of the method performance on RGB and thermal sequences, respectively. According to tables, the pixel-wise water segmentation recall is 95.23% on average for the RGB images and 94.29% on average for the thermal images, even in the presence of islands or other complex shoreline shapes. The measured average precisions are 93.88% and 95.41% for the RGB and thermal datasets, respectively.

As mentioned above, we propose to obtain the water map by applying the water detection method for T times in a sequence and by calculating the average over the resulting outputs, to remove irrelevant regions and decrease the water pixel miss rate. Then we validate this temporal averaging technique on a few available sequences that are of sufficient length for serious testing, applying different number of T iterations and time-intervals (number of frames skipped between iteration number T), which are specified in Tables 1 and 2 by columns with values T and Interval, respectively. Fig. 6, illustrates an example of such a case, where the method is applied with $T = 8$ and $Interval = 200$ on a test sequence captured from the Rhine river in Rotterdam harbor, The Netherlands. In this figure, in addition to the original frame and the final maritime region, three segmentation outputs along with the mean frame are depicted. The presented figure and the table data show that the method extracts the maritime region quite well.

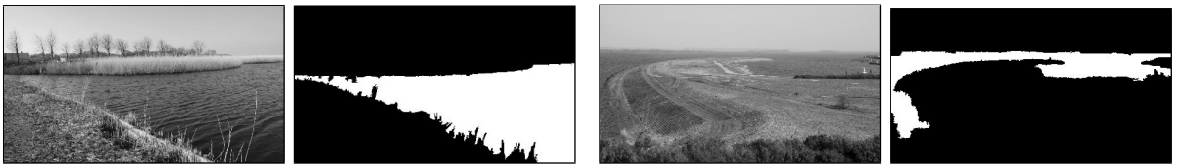
The validation results portrayed by Tables 1 and 2 reveal that the method provides high robustness in water region segmentation even in the presence of complex-shaped shorelines and islands. However, in our sequences there are scenes where the method does not properly extract the water regions. These happen especially where the flicker amount of energy histograms of water pixels drops due to absence of minimal dynamicity in a water part. Fig. 4d illustrates an example case where the missing water pixels are detected as a background, because those pixels are located at a very far distance and contain low dynamics. Additionally, in some other cases, there are regions belonging to non-water objects having temporary dynamic behavior (e.g. windblown veg-

Table 1: Quantitative analysis of proposed method on the RGB videos.

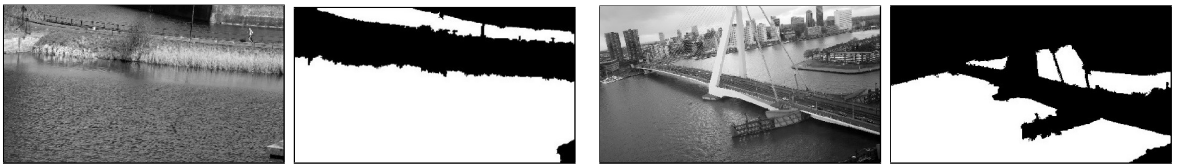
	Recall	Precision	*T*	Interval
Seq. 1	97.65	99.96	1	-
Seq. 2	98.20	99.92	1	-
Seq. 3	96.03	99.80	1	-
Seq. 4	95.13	99.37	1	-
Seq. 5	99.09	97.56	1	-
Seq. 6	94.10	92.54	1	-
Seq. 7	96.13	99.16	1	-
Seq. 8	93.29	98.53	1	-
Seq. 9	99.75	98.97	1	-
Seq. 10	91.18	98.91	1	-
Seq. 11	98.35	99.96	1	-
Seq. 12	97.27	99.19	1	-
Seq. 13	96.55	99.98	1	-
Seq. 14	98.32	54.36	1	-
Seq. 15	93.17	46.74	1	-
Seq. 16	83.60	85.40	5	500
Seq. 17	88.75	99.52	5	500
Seq. 18	98.06	99.20	5	300
Seq. 19	91.46	97.53	10	500
Seq. 20	98.45	99.97	10	200
Seq. 21	92.28	99.06	10	200
Seq. 22	98.29	99.67	8	200
Average	95.23	93.88	-	-

Table 2: Quantitative analysis of proposed method on the thermal videos.

	Recall	Precision	*T*	Interval
Seq. 23	96.78	85.26	1	-
Seq. 24	95.41	99.94	1	-
Seq. 25	92.95	100	1	-
Seq. 26	95.31	100	1	-
Seq. 27	93.73	95.86	1	-
Seq. 28	91.11	97.98	1	-
Seq. 29	96.97	100	1	-
Seq. 30	82.78	100	1	-
Seq. 31	88	99.97	1	-
Seq. 32	97.04	100	1	-
Seq. 33	96.84	98.68	1	-
Seq. 34	95.61	98.23	1	-
Seq. 35	97.25	99.58	1	-
Seq. 36	94.92	100	10	500
Seq. 37	95.56	81.39	10	500
Seq. 38	98.41	66.44	10	500
Seq. 39	94.32	98.71	10	600
Average	94.29	95.41	-	-



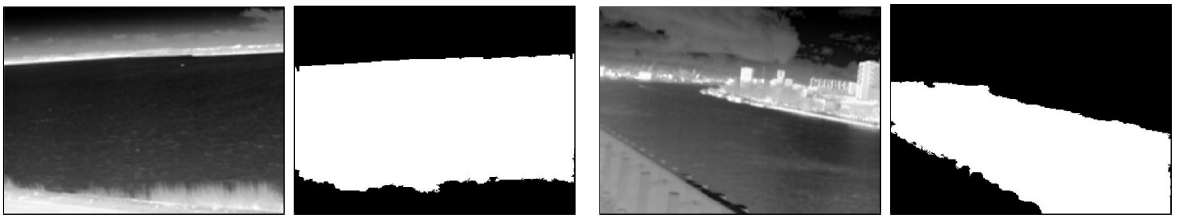
(a) **Seq. 10**, Original Frame/Extracted Water Region (b) **Seq. 16**, Original Frame/Extracted Water Region



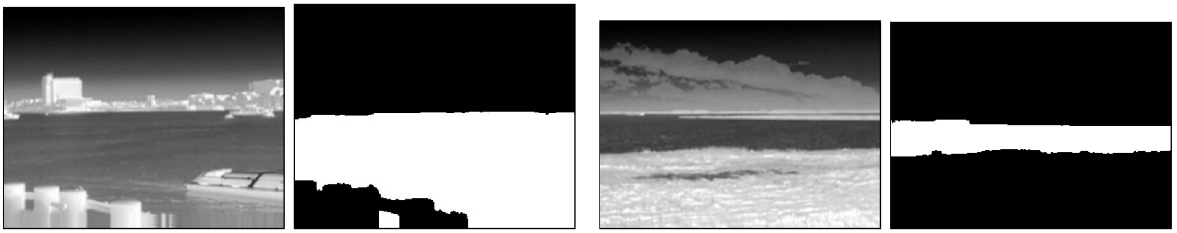
(c) **Seq. 21**, Original Frame/Extracted Water Region (d) **Seq. 17**, Original Frame/Extracted Water Region

(e) **Seq. 18**, Original Frame/Extracted Water Region (f) **Seq. 14**, Original Frame/Extracted Water Region

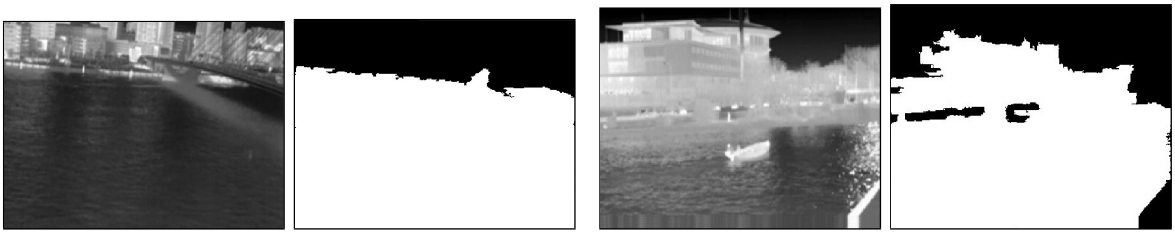
Figure 4: Water region extraction on 6 RGB sequences. In each case from left to right: original frame, and the extracted water region.



(a) **Seq. 24**, Original Frame/Extracted Water Region (b) **Seq. 34**, Original Frame/Extracted Water Region



(c) **Seq. 36**, Original Frame/Extracted Water Region (d) **Seq. 8**, Original Frame/Extracted Water Region



(e) **Seq. 13**, Original Frame/Extracted Water Region (f) **Seq. 38**, Original Frame/Extracted Water Region

Figure 5: Water region extraction on 6 thermal sequences. In each case from left to right: original frame, and the extracted water region.

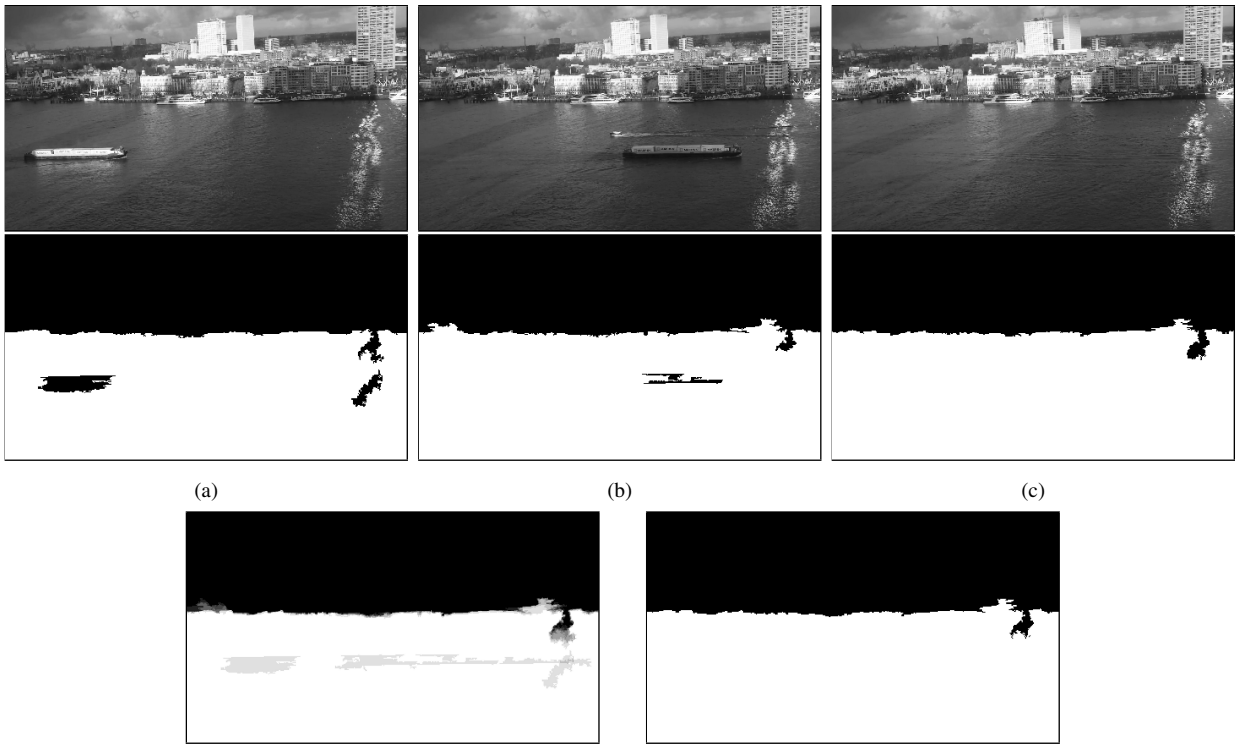


Figure 6: Temporal averaging technique solving water occlusions made by moving vessels. The first row illustrates three images of the Sequence 22. The second row shows the corresponding segmentation outputs. The third row depicts the mean frame resulting from the temporal averaging technique and the final extracted water region.

etation, building under light variations or moving clouds) which are segmented as water. Fig. 5f depicts such cases, where the sequence is captured during nighttime and lots of light variations occur on buildings and streets. In this case, a large irrelevant part of the scene is detected as water. Although such regions are often removed by the presented temporal averaging technique, for remaining cases we plan to combine other features (e.g. artefact structures by HOG features) to define stronger criteria, preventing such regions from incorrect labeling.

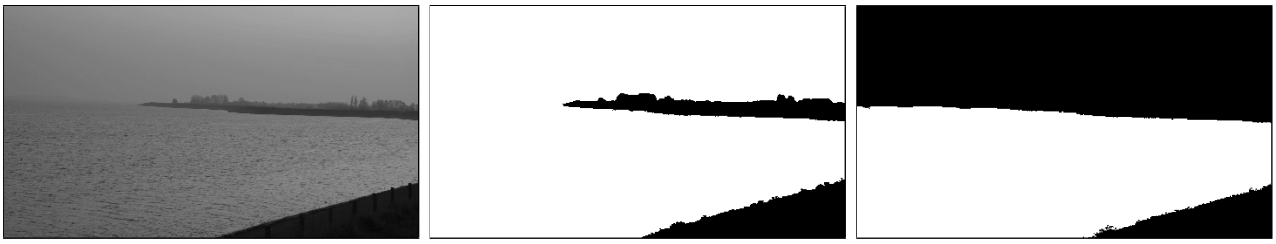
It is important to mention that the method incorporates eight thresholds for decision making. As a result, the method highly depends on the threshold values, which makes it hard to tune. For instance, there is an overall trade-off between thresholds which affect the size of initial groups (i.e. the thresholds of the mean shift smoothing and the ROLA grouping blocks) and thresholds determining if a group should be labeled as a water region. By decreasing the grouping thresholds, the size of generated groups increase and more pixels are included into one larger cluster. In this clustering process, non-relevant pixels may join a cluster while including water pixels as well. This leads to labeling of irrelevant regions as water in the identification step (independent of identification threshold values). Besides this, increasing the grouping threshold values may lead to smaller initial clusters. As a result, the water pixels which do not contain noticeable flicker amount in their energy histogram may gather in a group separated from wavy water pixels. Consequently, quiet water pixels would have less chance to be identified as a water region against the water identification thresholds.

In Fig. 7, we illustrate one example to make the threshold-dependency problem more evident. Fig. 7a illustrates the original frame. In Fig. 7b and 7c, the detected water regions with two different thresholds are depicted. The detected water region in Fig. 7b includes the sky as well. This happens due to the dark scene, where both the sky and the water do not have significant difference between their borders. Additionally, clustering thresholds have low values for this case, such that the sky pixels have been grouped together with the water part. At the identification step, the method labels all the sky pixels as water. But in Fig. 7c, we have increased the grouping thresholds, such that pixels are grouped with more similar pixels. Consequently, the method clusters the water and sky separately.

Conclusions

Despite the importance of ROI detection for maritime surveillance methods (i.e. vessel detection, tracking and classification), state-of-the-art methods lack an accurate water extraction in a pre-processing stage. Although a few algorithms exist which extract the water regions using classifiers and/or features like color, texture and spatiotemporal statistics of pixel groups, these methods were only evaluated on data sets with simple scenes.

In this paper we have therefore proposed and validated an algorithm that is robust in water extraction from complex scenes containing various scenes on rivers, channels, lakes and sea sides, having shorelines with curved shapes, islands, bridges, and wind-blown vegetation. Besides this, the validation datasets were captured from different camera heights, during daytime and night-



(a) Main Frame (b) Extracted Maritime Region (c) Maritime Region With Other Thresholds

Figure 7: Threshold dependency. From left to right: (a) original frame from Sequence 14, (b) representative water region extracted with the thresholds specified in the paper, and (c) water region extracted with higher grouping thresholds.

time and under variable weather conditions (e.g. sunshine, clouds, wind, rain).

The proposed algorithm is based on exploiting spatiotemporally-oriented energy features which provide a rich source of information when they are jointly exploited. In the proposal, up to 8 features are used, involving both static directions and motion directions and a lack-of-structure feature. The mean shift algorithm smooths the outcomes and clusters the scene into coherent regions. The third part of our proposed system is a raster-order based labeling algorithm (ROLA) to assign the same labels to clusters with corresponding properties.

Another important contribution is the method ability to extract water regions in thermal images. Thermal sensors provide a beneficial modality for the maritime surveillance tasks, since they are able to capture data even during nights and foggy situations. Due to the low resolution of thermal images, water extraction becomes an even more challenging task. To our best knowledge, methods on water extraction from thermal data were not reported in the literature yet. The new presented method features surprisingly high recall (95.23 - 94.29% on average) and precision (93.88 - 95.41% on average).

References

[1] Chen, Y., S. H. Wang, G. P. Wang, W. L. Chen, and J. L. Wu, Automatic detection of IR ship on the basis of variance statistic, Proc. MCCA, pg. 273. (2014).

[2] Tang, Da, Gang Sun, Ding-he Wang, Zhao-dong Niu, and Zeng-ping Chen, Research on infrared ship detection method in sea-sky background, Proc. InISPDI, pg. 89072H. (2013).

[3] Teutsch, Michael, and Wolfgang Krger, Classification of small boats in infrared images for maritime surveillance, Proc. WSS, pg. 1. (2010).

[4] Szpak, Zygmunt L., and Jules R. Tapamo, Maritime surveillance: Tracking ships inside a dynamic background using a fast level-set, J. Expert systems with applications, 38(6), 6669 (2011).

[5] Wei, Hai, Hieu Nguyen, Prakash Ramu, Chaitanya Raju, Xiaoqing Liu, and Jacob Yadegar, Automated intelligent video surveillance system for ships, Proc. SPIE, pg. 73061N. (2009).

[6] Pires, Nuno, Jonathan Guinet, and Elodie Dusch, ASV: an innovative automatic system for maritime surveillance, J. Navigation, 58(232), 1 (2010).

[7] Fefilatyev, Sergiy, Algorithms for Visual Maritime Surveillance with Rapidly Moving Camera, 2012.

[8] Wu, Fang, Chengfei Zhu, and Wenfang Xue, Detect ships using saliency in infrared images with sea-sky background, Proc. ICDIP, pg. 96310A. (2015).

[9] Liu, Zhaoying, Changming Sun, Xiangzhi Bai, and Fugen Zhou, Infrared ship target image smoothing based on adaptive mean shift, Proc. DICTA, pg. 1. (2014).

[10] Frost, Duncan, and Jules-Raymond Tapamo, Detection and tracking of moving objects in a maritime environment using level set with shape priors, J. Image and Video Processing, 1, 1 (2013).

[11] Saghafi, M., S. Javadein, Seyed Majid, and H. Noorhosseini, Robust Ship Detection and Tracking Using Modified ViBe and Backwash Cancellation Algorithm, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2014, pg. 117.

[12] Hu, Wu-Chih, Ching-Yu Yang, and Deng-Yuan Huang, Robust real-time ship detection and tracking for visual surveillance of cage aquaculture, J. Visual Communication and Image Representation, 22(6), 543 (2011).

[13] Arshad, Nasim, Kwang-Seok Moon, and Jong-Nam Kim, Multiple ship detection and tracking using background registration and morphological operations, Proc. SIP and MulGraB, pg. 121. (2010).

[14] Gupta, Kalyan Moy, David W. Aha, Ralph Hartley, and Philip G. Moore, Adaptive maritime video surveillance, Proc. SPIE, pg. 734609. (2009).

[15] Bao, X., S. Zinger, and R. G. J. Wijnhoven, Peter H.N. de With, Water region detection supporting ship traffic analysis in port surveillance, Proc. ACIVS, (2012).

[16] Rankin, A., and L. Matthies, Daytime water detection and localization for unmanned ground vehicle autonomous navigation, Proc. ASC, pg. 1. (2006).

[17] Rankin, A. L., L. H. Matthies, and A. Huertas, Daytime water detection by fusing multiple cues for autonomous off-road navigation, Transformational Science and Technology for the Current and Future Force, 2004, pg. 177.

[18] Mettes, Pascal, Robby T. Tan, and Remco C. Veltkamp, Water detection through spatio-temporal invariant descriptors, J. Computer Vision and Image Understanding, 1 (2016).

[19] R. P. Wildes and J. R. Bergen, Qualitative spatiotemporal analysis using an oriented energy representation, Proc. ECCV, pg. 768. (2000).

[20] W. T. Freeman and E. H. Adelson, The design and use of steerable filters, J. IEEE Transactions on Pattern analysis and machine intelligence, 13(9), 891 (1991).

[21] K. G. Derpanis and J. M. Gryn, Three-dimensional nth derivative of Gaussian separable steerable filters, Proc. ICIP, pg. 553. (2005).

[22] Cannons, Kevin J., and Richard P. Wildes, The applicability of spatiotemporal oriented energy features to region tracking, J. IEEE transactions on pattern analysis and machine intelligence, 36(4), 784 (2014).

[23] Cannons, Kevin J., Jacob M. Gryn, and Richard P. Wildes, Visual

tracking using a pixelwise spatiotemporal oriented energy representation, Proc. ECCV, pg. 511. (2010).
[24] Derpanis, Konstantinos G., and Richard P. Wildes, Dynamic texture recognition based on distributions of spacetime oriented structure, Proc. CVPR, pg. 191. (2010).
[25] Derpanis, Konstantinos G., and Richard P. Wildes, Early spatiotemporal grouping with a distributed oriented energy representation, Proc. CVPR, pg. 232. (2009).
[26] K. G. Derpanis, M. Sizintsev, K. Cannons and R. P. Wildes, Efficient action spotting based on a spacetime oriented structure representation, Proc. CVPR, (2010).
[27] D. Comaniciu and P. Meer, Mean shift: A robust approach toward feature space analysis. J. IEEE transactions on pattern analysis and machine intelligence, 24, 603. (2002).
[28] Derpanis, Konstantinos G., The bhattacharyya measure, Mendeley Computer 1(4), 1990 (2008).

Author Biography

Amir Ghahremani received his BSc. degree in Electrical Engineering with emphasis on telecommunication from Azad University, Urmia, Iran, in 2009. This was followed by a MSc. degree in Electrical Engineering with emphasis on Electronics at the Khajeh Nasir Toosi University of Technology, Tehran, Iran, in 2014. Since 2015, he is working as a PhD at the Technical University of Eindhoven (TU/e), the Netherlands. His research interests include computer vision, semantic content analysis, and machine learning.

Egor Bondarev received his MSc degree in robotics and informatics at the State Polytechnic University, Belarus Republic, in 1997. In 2009 he has obtained his PhD degree in Computer Science Department at Eindhoven University of Technology (TU/e), The Netherlands in the domain of performance predictions of real-time component-based systems on multiprocessor architectures. Currently, he is an Assistant Professor at the Video Coding Architectures (VCA) group, TU/e, focusing on such research areas as surveillance by multi-modal sensor fusion and photorealistic 3D reconstruction. He is active in TUE education agenda, giving courses on Computer Architecture and lectures on Advanced Embedded Vision for bachelor and master students. Besides this, his research results in numerous journal and IEEE conference publications.

Peter H. N. de With (MSc. EE) received his PhD degree from University of Technology Delft, The Netherlands. After positions at Philips Research, University Mannheim, LogicaCMG and CycloMedia, he became full professor at Eindhoven University of Technology. He is an (inter-)national expert in surveillance for safety/security and was involved in multiple EU projects on video surveillance analysis with the Harbor of Rotterdam, Dutch Defense, Bosch Security, TKH-Security, ViNotion, etc. He is board member of the Dutch Institute of Safety and Security (DITSS) and R&D advisor to multiple companies. De With is IEEE Fellow, has (co-)authored over 350 papers on video analysis, systems and architectures, and received multiple awards of the IEEE, VCIP, and EURASIP.