Context modeling combined with motion analysis for moving ship detection in port surveillance

Journal of Electronic Imaging

SPIEDigitalLibrary.org/jei

Context modeling combined with motion analysis for moving ship detection in port surveillance

Xinfeng Bao
Solmaz Javanbakhti
Svitlana Zinger
Rob Wijnhoven
Peter H. N. de With

Context modeling combined with motion analysis for moving ship detection in port surveillance

Xinfeng Bao
Solmaz Javanbakhti
Svitlana Zinger
Eindhoven University of Technology
Video Coding and Architectures Research Group (SPS-VCA)
Electrical Engineering Faculty
P.O. Box 513
5600 MB Eindhoven, The Netherlands
E-mail: x.f.bao@tue.nl

Rob Wijnhoven
ViNotion B.V.
Horsten 1, 5600 CH Eindhoven, The Netherlands

Peter H. N. de With
Eindhoven University of Technology
Video Coding and Architectures Research Group (SPS-VCA)
Electrical Engineering Faculty
P.O. Box 513
5600 MB Eindhoven, The Netherlands

Abstract. *In port surveillance, video-based monitoring is a valuable supplement to a radar system by helping to detect smaller ships in the shadow of a larger ship and with the possibility to detect nonmetal ships. Therefore, automatic video-based ship detection is an important research area for security control in port regions. An approach that automatically detects moving ships in port surveillance videos with robustness for occlusions is presented. In our approach, important elements from the visual, spatial, and temporal features of the scene are used to create a model of the contextual information and perform a motion saliency analysis. We model the context of the scene by first segmenting the video frame and contextually labeling the segments, such as water, vegetation, etc. Then, based on the assumption that each object has its own motion, labeled segments are merged into individual semantic regions even when occlusions occur. The context is finally modeled to help locating the candidate ships by exploring semantic relations between ships and context, spatial adjacency and size constraints of different regions. Additionally, we assume that the ship moves with a significant speed compared to its surroundings. As a result, ships are detected by checking motion saliency for candidate ships according to the predefined criteria. We compare this approach with the conventional technique for object classification based on support vector machine. Experiments are carried out with real-life surveillance videos, where the obtained results outperform two recent algorithms and show the accuracy and robustness of the proposed ship detection approach. The inherent simplicity of our algorithmic subsystems enables real-time operation of our proposal in embedded video surveillance, such as port surveillance systems based on moving, nonstatic cameras.* © *2013 SPIE and IS&T [DOI: 10.1117/1.JEI.22.4.041114]*

Paper 13204SSP received Apr. 16, 2013; revised manuscript received Jul. 4, 2013; accepted for publication Jul. 29, 2013; published online Aug. 30, 2013.

1 Introduction

In port areas, various hazardous scenarios occur, which are caused by heavy traffic conditions and the mixing of large sea ships with local smaller vessels. In particular, dangerous situations can occur when small ships travel in the radar shadow of large ships, so that they become invisible for the radar system and the harbor management. Visual surveillance is a possibility, but because of the large diversity of ships' functionalities and shapes, human visual inspection is highly laborious and error-prone. Automatic ship detection is an attractive research topic in the field of port surveillance which can nurture various applications such as vessel traffic monitoring, ship identity management, and smuggling prevention.

To complement the deficiencies of a radar system, different port surveillance technologies are explored. Although satellite imagery has sufficient resolution to perform ship detection, those systems are highly sensitive to noise and not efficient in detecting small ships. Moreover, limited by the low frequency of satellite revisit, those systems cannot perform 24/7 monitoring on the port area.[1–4] Some systems are based on infrared imagery,[5–7] in which the acquired images are suitable for automatic detection and recognition. Another benefit is that those systems are able to work in extreme conditions such as hazy weather or lack of visible lights. However, infrared systems are costly and prone to the damages caused by bright lights.

Camera-based ship detection is another attractive option due to its low cost and the ease of management both in installation and maintenance. Although video-based techniques

Fig. 1 Examples of dynamic/complex surroundings and various types of ships in port surveillance videos.

are broadly explored for vehicle detection along roads, video analysis for ship detection still remains as a domain of active research. Ships traveling in highly dynamic water regions and complex surroundings largely limit the usage of conventional background modeling, which is typically applied in this kind of research. Furthermore, highly variable ship appearances intrinsically bring in difficulties for constructing robust matching templates. Figure 1 shows a few examples of different appearances of surroundings and the large variation of ships in port surveillance videos.

In the last decade, a number of papers discussed the promising usage of video cameras to perform the ship detection task. Based on the state-of-the-art literature, there are mainly two types of techniques used for ship detection: background estimation and appearance modeling. For background estimation, some techniques[8–10] aim at first detecting the horizon line in the frame and separating the ships from the modeled sky or water using image registration and subtraction. Socek et al.[11] proposed a more general approach where they cluster color features through segmentation and feed them to a background registration. Using the modeled background, they detect the foreground. Arshad et al.[12] use edge information instead of color and use them in morphological operations based on background modeling. Generally, these methods try to solve ship detection problems by modeling the background. However, in real-life surveillance videos, the background has large variations due to the dynamic surroundings in port regions, such as illumination changes and scintillation, which easily lead to failures of such approaches. Although those failures can be reduced by creating dynamic background models, the high complexity makes those techniques inadequate for real-time applications. Moreover, systems using background estimation either work for fixed camera or require a setup time to model the background in case of change of camera position. All these approaches require the camera to be static during the normal operation of the algorithm and do not operate when the camera moves to a different position. We target a system where the surveillance continues even during the camera movement.

As for the appearance modeling approaches, they mostly rely on the local features. Although local properties are fundamental in object detection research, its performance is easily affected by the light variations and dependent on the complexity of the object appearances. On one hand, objects with specific regional appearances, like smoothness and position in the image, can be recognized as either part of sky or water using local features. On the other hand, objects with variable appearances and displacement through the image, such as ships, cannot be described by local features to construct robust appearance descriptors. Sullivan and Shah[13] created complete descriptors by training a set of filters for each vessel class. By applying each template to the videos and analyzing the outputs of cross-correlation in the frequency domain, the system can locate the ships in the image. However, their algorithm tends to miss a target if the appearances of the ship differ from the pretrained templates. Wijnhoven et al.[14] make efforts to utilize local descriptors for representative parts of ships instead of modeling the complete ship appearances. They build a cabin detector based on a histogram of oriented gradients (HOG)[15] and classify the resulting patterns. However, the simplified local descriptors are hardly distinctive from other highly textured patches in the image, such as vegetation. Moreover, the algorithm fails to work on the ships without cabins.

We envision two approaches for improving the reliability of ship detection. The first way is to design a complete but more complicated appearance model using local features. The model aims at solving the above-mentioned limitations of ship detection, especially when the targeted objects are noisy or partly occluded. However, the intrinsic complexity of such method limits its usage in real-time applications. The second approach follows the concept of context-based object detection in recent research.[16,17] The performance of local descriptors can be significantly enhanced when combined with contextual information. The reason for it is a spatial and temporal co-occurrence consistency between different objects or between an object and its surroundings. For example, a moving flag will remain in the vicinity of its surroundings. It has been shown that in object detection and recognition tasks, this consistency can provide a rich source of information to reduce ambiguities of local appearances of different objects,[16,18] which leads to an improved reliability and probability of detection.

In our earlier work,[19,20] we presented a preliminary framework of context-based ship detection by extracting the water region as contextual information. Although the context is not fully modeled because only water regions are labeled, the performances of those approaches are promising for the context-based detection.

In this paper, we present a ship detection system based on moving camera, where the system can detect ships without initiation when the field of view in the system changes. We further develop a context-based method to perform automatic moving ship detection, which completes the context model explored in our previous work.[19,20] Additionally, the

temporal properties of moving ships are exploited to fuse the visual, spatial, and temporal features in a single framework. Although some previous work also extracts supporting information to facilitate ship detection, such as modeled water region,[9,21] the water region is only used as a background model and modeled at the pixel level. This approach is not suitable for a moving camera-based system. In our improved approach, we consider more advanced usage of such supporting information. We extend water region extraction to region labeling and explore the relations between ship and those labeled regions in a context model. In the meantime, the region appearances are modeled at the region level, which achieves higher robustness. Temporal features are also explored in ship detection.[9,10,21,22] However, a temporal feature approach either relies on multisource images or a complete tracking system, both of which have a high complexity for real-time applications.

Generally, our approach is based on the following two observations in the scenario of port surveillance: (1) ships can only travel within the water region and (2) each ship has a particular motion that distinguishes itself from other ships and from the surroundings. Concerning the motion characteristics of ships, we note that each ship has its own motion pattern and its motion is more significant than the motion within the local background. Based on these observations, we first explore the context information using region labeling and motion similarity analysis to reliably derive the positions of ships in the scene. We consider three aspects to model the context information:[23]

1. Semantic context—presence of an object indicates other objects' existence in a port scene.
2. Spatial context (position)—natural or logical geometric placement of different objects in a port scene.
3. Scale context (size)—size dependence of different objects or constraints between an object and surroundings.

After the context modeling, the region-level motion of presumed ships and the corresponding local background are analyzed to detect the ships. The major advantage of our approach is that it combines two techniques: context modeling with motion similarity and a separate stage with motion saliency analysis. Both techniques are designed to have moderate complexity, enabling a real-time implementation for embedded port surveillance while performing reliable ship detection. The two proposed algorithmic subsystems are also designed to operate on videos obtained by moving cameras. Thus, it requires no background substraction techniques and no prior knowledge of ship appearances for a higher robustness in real-life applications. Furthermore, it can handle occlusions between ships as well as clutters between ships and vegetation.

This paper is organized as follows. In Sec. 2, we give a short and clear overview of our ship detection approach. In Sec. 3, we mainly focus on describing the context modeling, which includes region labeling for semantic understanding of the video, the segment merging into semantic regions based on motion similarity, and the context extraction based on these regions. In Sec. 4, we discuss the analysis of motion saliency based on the extracted context for moving ship detection. In Sec. 5, we present the experimental results for both region labeling and ship detection. Additionally, we compare this approach with the ship detection based on HOG[15] and support vector machine (SVM)[14,19,24] and an initial version of the concept discussed in this paper.[20] Finally, Sec. 6 presents conclusions and discusses future work.

2 Overview of Our Approach

We consider two frameworks to perform context-based ship detection (see Figs. 2 and 3). The first framework (framework A) employs parallel processing for context modeling and ship detection, where the two results are fused with a verification stage. The false alarms can be reduced when the detection adequately benefits from the extracted context information. To enable parallel processing, the ship detection should be independent of the context modeling, which emphasizes the use of a more complex and robust appearance model for ships, according to the discussion in Sec. 1. The second framework (framework B) processes the context modeling and ship detection in a sequential strategy, as shown in Fig. 3. The extracted context in this framework not only provides additional information for ship detection, but also creates the initial detection results, enabling a complexity reduction of ship detection. The sequential framework exempts the approach from solving the problems of ship detection algorithms where no context is applied. Furthermore, the simplified ship detection enables real-time implementation for port surveillance.

We further explore framework B to accomplish the ship detection through a two-stage sequential approach, combining the context modeling and motion analysis. At the first stage, we aim at modeling the context for a better understanding of the port scenario. A graph-based segmentation[25] is employed to divide a video frame into segments. The object-centric region labeling is then employed to classify those segments into three classes: water, vegetation, and unknown. The labeled segments are then used to analyze motion similarity employing statistical region merging (SRM).[26] Adjacent segments with the same labels and statistically similar motion are merged into semantic regions, through which occluded regions are also separated from each other. These regions are analyzed based on semantic, spatial, and scale constraints to build the context model, which provides knowledge of locations of candidate ships. At the second stage, based on the common understanding that ships should have more significant motion, the regions with salient motions are detected as moving ships. This ship detection approach is a first initial step; when more

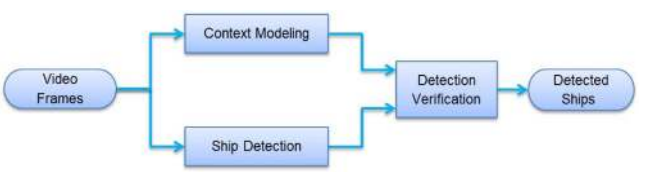


Fig. 2 Framework A: ship detection with parallel processing of context modeling and ship detection.

Fig. 3 Framework B: ship detection with sequential processing of context modeling and ship detection.

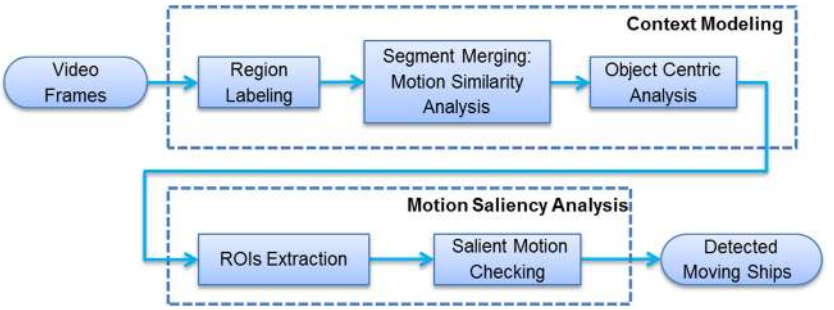


Fig. 4 Flow chart of the moving ship detection based on context modeling and motion saliency analysis.

knowledge about the ships is required, additional features can be added. In our research, salient motion is defined based on a set of criteria to distinguish it from other types of motion, such as the scintillation/ripples of the water surface and the wind-based motion of vegetation. Our approach is depicted in Fig. 4, which we will further discuss in detail in the following sections.

3 Context Modeling

In a context-based object detection and recognition task, there are basically two types of approaches to model the context.[17] The first is scene-centric, which models the context at the entire image level to generate a description of the scene, such as the theme or gist of the scene.[27,28] The scene-centric methods either model the context with statistics vectors of low-level visual features, such as pixel-based color, texture, and shape features, computed from the whole image or they group vocabularies like bag-of-words[29] with those features. The second approach is object-centric, which explores the relationships between different objects like the co-occurrence or the spatial replacement.[30–32] In our ship detection approach, the surveillance video already gives an implicit gist for the scene. Therefore, we focus on generating an object-centric context model that gives knowledge concerning the possible locations of ships. To model the context, an object-centric region labeling is performed followed by a motion analysis. Finally, an object-centric analysis of the semantic regions is conducted.

3.1 *Object-Centric Region Labeling*

As the demand for surveillance application grows, a huge amount of surveillance video is captured daily. The workload of the operators or analysts become bigger and bigger. An automated video processing tool is needed to reduce the workload of these operators. This desired video processing tool should automatically extract information that is easy to summarize or analyze by an end user.[33] Region labeling in a video will contribute to the semantic understanding of that video, such as a better understanding of the natural surroundings and their arrangement in a port scene. In this section, we present our fast region labeling approach. We propose a region labeling system based on the observation that each region is more likely to be found at a specific vertical position. A region corresponding to a specific semantic meaning normally covers a certain part of the color space and has a distinct texture. We aim at classifying n types of regions such as sky, vegetation, water, etc. Our algorithm contains three stages, as depicted in Fig. 5 and given below.

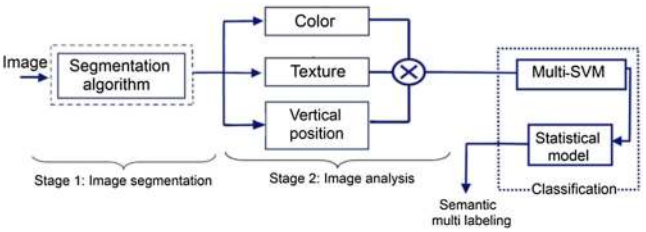


Fig. 5 Block diagram of our region labeling approach.

- *Stage 1: Context/region segmentation*: In our system, the basic idea is to combine the image segmentation with the region classification technique. Instead of labeling each region directly at the pixel level, we first divide the image into several regions with uniform color (or texture), which sufficiently considers additional assumptions on the color continuity discussed later.
- *Stage 2: Context/region analysis*: In this step, global and local features of each segmented region are extracted.
- *Stage 3: Context/object classification*: This is divided into two aspects as follows:
  - *Multi-SVM (one versus all)*: For each class of regions, we use an off-line separately trained SVM (a binary classifier). The SVM classification provides a better performance compared with the other approaches,[34] and the method is generic and can be reused for different types of objects. Therefore, it is further explored for region labeling in this paper.
  - *Statistical model process*: As postprocessing, we compare the percentage of labeled pixels with a predetermined threshold T for each region. We assign a label to a region for which the percentage of positively classified pixels is $\geq T$.

3.1.1 *Context/region image segmentation*

We adopt an efficient graph-based segmentation from Ref. 25 as preprocessing in our region labeling to achieve two objectives: (1) distinguish each region from other objects while preserving the overall characterization of the region itself and (2) perform fast segmentation to support a real-time application in surveillance systems.[35] The basic idea

of the graph-based method is that pixels within one region are closer in color space than pixels from different regions.

We define the segmentation stage more formally. For each pair of neighboring pixels i, j, there is an edge with an Euclidean weight $w_{i,j}$, which is specified by

$$w_{i,j} = \sqrt{(R_i - R_j)^2 + (G_i - G_j)^2 + (B_i - B_j)^2}, \quad (1)$$

where R_i, G_i, B_i are the RGB color values of the pixel i (or j). Based on a normalized $w_{i,j}$, a threshold is defined depending on the size of the region, in which we assume that large regions should be given a higher tolerance. If a region is large, it tends to incorporate more neighboring pixels. Three parameters are considered: the standard deviation σ, initial inner threshold κ, and minimum region size $S_{\min}$. The image is first blurred using a Gaussian filter, and σ is the standard deviation of the Gaussian filter. To determine if two regions should merge, intraregion weight (W_A) of region A is defined as the maximum edge weight within the region. Initially, each pixel is regarded as a region, and the initial threshold τ for each region is set to $\tau_A = \kappa/|A|$, where $|A|$ is the pixel size of region A. Note that $|A| = 1$ if region A is 1 pixel and κ is a constant parameter controlling the merging, such that a larger value results in larger segments.

Two weights are further defined: (1) inter-region weight of a pair of regions A and B [$W_{m-\text{inter}}(A, B)$], meaning the minimum edge weight between those two regions and (2) the minimum of intraregion weights of the involved regions A and B [$W_{m-\text{intra}}(A, B)$]. This weight now becomes

$$W_{m-\text{intra}}(A, B) = \min(W_A + \tau_A, W_B + \tau_B). \quad (2)$$

The two regions A and B are merged into a new region if it satisfies the following condition:

$$W_{m-\text{inter}}(A, B) < W_{m-\text{intra}}(A, B). \quad (3)$$

If the merge occurs, it is evident that the weight for the merged region ($A \cup B$) now becomes identical to $W_{(A\cup B)} = W_{m-\text{inter}}(A, B) + \tau_{(A\cup B)}$. Note that very small regions are merged based on the minimum region size, even if the merging criterion is not satisfied.

The above concept is incorporated in Algorithm 1.

Algorithm 1 Our graph-based segmentation approach in pseudo-language.

```
Initialization: regard each node as a region and sort edges in a
nondecreasing order of weights w

for each region do

  Extract the two nodes A and B it connects;

  if A and B are different regions and W_{m-intra}(A, B) is computed
  as in Eq. (2) then

    if W_{m-inter}(A, B) < W_{m-intra}(A, B) then

      Join A and B as a new region; Update the weight of the
      new region;

    end

  end

end
```

3.1.2 *Context/region analysis: feature extraction*

This involves the second stage in our region labeling system. Prior to training a reliable and robust SVM classifier, it can be sufficient to use only local features such as color and texture. However, when classes have similar characteristics (overlapping classes), complications arise. These complications can be solved by using spatial context as an additional feature. This context involves and exploits the vertical position of the regions in the image, e.g., the water is typically at the bottom of the image. Later, we will also employ vegetation detection to find possible occlusions, so that we use a generalized concept of region labeling. Summarizing, we combine the locally calculated pixel-based features and the region-based features to achieve a more reliable region labeling approach. We have trained and tested two different datasets with different scenes and the results are satisfying. This section presents all features used for analyzing the images.

Pixel-based features. (1) Color may be one of the most straightforward features utilized by humans for visual recognition and discrimination.[36] Here, we use the RGB color space. (2) Texture takes into account the local neighborhood variation and a better classification is achieved when texture information is included in the analysis.[37] Gabor features are widely applied to computer vision and image analysis. In addition to accurate time-frequency location, they also provide robustness against varying brightness and contrast of images.[38] Based on these properties, we apply here a group of Gabor filters with three scales and six orientations.

Region-based features. We propose two different region-based features based on the vertical position of each region, i.e., water and vegetation. To compose the feature vector, we combine the above pixel-based features with one of the following two region-based features. If the feature vector is composed of pixel-based features and the spatial context (SC), we call the approach a gravity model. If the feature vector is composed of pixel-based features and global region statistics, the approach is called a statistics-based model.

- SC helps to perform accurate region labeling.[18] For each pixel (i, j), we calculate its normalized vertical position $\text{SC}_{ij} = i/n$, where i is the row number in the image and j the column number. Each region consists of n rows. We call this method a gravity model. We will use this concept further in this paper.
- Global region statistics model the vertical location of regions in the set of images. The statistics are computed as follows. Let us assume that we have M regions of a particular type, for example, sky, in the training set of images. For each region, we calculate mean values μ_k $(k = 1, \ldots, M)$ of the vertical positions of its pixels. We also calculate the standard deviation σ_k of the vertical pixel positions for each region. Then we take minimum and maximum values for all means and standard deviations for this region type:

$\mu_{\min} = \min(\mu_1, \ldots, \mu_M)$, $\mu_{\max} = \max(\mu_1, \ldots, \mu_M)$, $\sigma_{\min} = \min(\sigma_1, \ldots, \sigma_M)$ and $\sigma_{\max} = \max(\sigma_1, \ldots, \sigma_M)$. In this way, we obtain intervals for mean and variance for the region. We assume that the mean value of vertical pixel positions lies in the interval $(\mu_{\min}, \mu_{\max})$ and the standard deviation in $(\sigma_{\min}, \sigma_{\max})$. We find these intervals for each of the n region types described in this paper. We call this method a statistics-based model. We employ this method for comparison with the above gravity model.

3.1.3 *Context region/object classification approaches*

After segmenting and analyzing the image, we proceed to obtain the labeling results. The labeling is performed by a classification system based on an off-line trained SVM. Here, we present two approaches for region classification.

Fast classification using the gravity model. In the first region classification approach, color, texture, and spatial context are used to train the SVM (one versus all) for each class of regions separately, i.e., an individual binary SVM (unitary-category classifier) is trained for each type of region. Our fast unitary-category classification is described in the inner part of Algorithm 2.

For multicategory labeling, we assign to each segment one of the n labels such as sky, vegetation, water, etc. To this end, we classify each segment by n specified SVMs using our unitary-category classification Algorithm 2 and obtain n values, indicating the percentages of positive pixels for each SVM as described in Eq. (4) below.

Algorithm 2 Our fast multicategory classification with embedded unitary-category algorithm (a binary classifier) in pseudo-language.

```
for n classes do
    Define the next class type;
    Set corresponding threshold T_eclass;
    for a segmented region do
        Randomly choose 100 pixels in this region and use the SVM
        classifier to label the pixels;
        Calculate the percentage of positive pixels in this region
        if the percentage of positive pixels exceeds the possibility
        threshold p then
            Set this region to positive
        End
        Compare results to class threshold T_eclass
        Label the current region
    End
end
```

$$P = [P_{\text{class}}], \qquad \text{class} \in \{1, 2, \ldots, n\}, \tag{4}$$

where numbers $1, \ldots, n$ represent n labels. Following this, we calculate the maximum percentage $M_P = \max(P)$ and then we compare it with the corresponding empirically determined threshold $T_{e_{\text{class}}}$. This threshold was tuned after processing of a broad random set of images from the same dataset. Finally, a segmented region is assigned to a particular class if its percentage is higher than the corresponding empirically determined threshold $M_P > T_{e_{\text{class}}}$. If it is not higher, we repeat this step for the second maximum, and so on.

Algorithm 2 illustrates our multicategory classification algorithm with the unitary-category algorithm embedded into it. The empirical threshold $T_{e_{\text{class}}}$ for each region has been found to be in the interval $(0.1, 0.8)$.

Classification using the statistics-based model. We propose a second region classification approach to classify images based on the position of each region using statistical information from the statistics model defined in Sec. 3.1.2. First, we apply the SVM with color and texture while using the same steps as in the first classification approach. Then, for assigning a label to a region, we check that both the following conditions are satisfied: (1) the percentage of positively classified pixels exceeds the threshold T_e (otherwise, they are labeled as "unknown") and (2) the mean and variance of the vertical positions of the pixels lie in the intervals defined in Sec. 3.1.2.

3.2 *Segment Merging*

In our ship detection, we semantically interpret a port area scene, where water and vegetation are the two dominant regions. To achieve that, the segments obtained in Sec. 3.1.1 are divided by the aforementioned object-centric region labeling into three classes: water, vegetation, and unknown. Afterward, we have to merge the labeled segments into semantic regions. Although merging based on labels and spatial adjacency is possible, the occluded objects with the same labels will be merged into one region, which will deteriorate the detection results. In order to avoid this, we consider motion similarity between these segments.[39] We assume that segments from the same object should have similar motion patterns, which distinguish them from other objects. Therefore, based on motion similarity, spatial adjacency, and labels, we design a more reliable merging process to group the segments while separating the occluded semantic regions.

We first calculate the pixel-wise motion for the image using optical flow.[40–42] Derived from the SRM algorithm,[26] we define a merging predicate $P(C_i, C_j)$ to determine whether two segments C_i and C_j are from the same statistical region ($i \neq j$). Instead of using color features as in Ref. 26, we reuse the criterion with motion features in segment level. We use an average flow vector, which is the average of flow vectors of all pixels in a segment, to represent the motion of a segment. Based on the segmentation results (Sec. 3.1.1), we create a motion map by calculating the values of magnitude MAG and angle ANG for each average flow vector. The values of MAG belong to the set $\{1, 2, \ldots, g_{\text{MAG}}\}$ and ANG to $\{1, 2, \ldots, g_{\text{ANG}}\}$ (here, the set sizes $g_{\text{MAG}} = 60$ and

$g_{\mathrm{ANG}} = 360$). Each segment in the motion map is assumed to be described by a set of distributions. In the motion map, the semantic regions representing objects should have a common homogeneity property in two ways: (1) In a certain statistical region, each statistical segment has the same expectation in both MAG and ANG. (2) For two adjacent statistical regions, the expectations differ from each other in either MAG or ANG values.

As SRM, we consider the merge of two segments only if they are spatially adjacent. Furthermore, consider there is a possibility that two adjacent segments from two different objects have similar motion, requiring the definition of constraints in our motion similarity analysis to prevent a merging of those segments. Since the region labeling (Sec. 3.1) provides the label information of each segment, we can use the information and impose a constraint in merging criteria that each statistical region should contain only segments with the same label. Suppose MAG and ANG are represented by a set of Q independent random variables and any possible sum of those variables belong to $\{1, 2, \ldots, g_{\mathrm{MAG}}\}$ and $\{1, 2, \ldots, g_{\mathrm{ANG}}\}$, respectively, the merging predicate can be defined as

$$P(C_i, C_j) = \begin{cases} \text{true} & \text{if } \forall\ k \in \{\mathrm{MAG}, \mathrm{ANG}\} \text{it holds that} \\ & (|\bar{C_j}(k) - \bar{C_i}(k)| \le \sqrt{b^2(C_i, k) + b^2(C_j, k)}) \wedge [L(C_i) = L(C_j)], \\ \text{false} & \text{otherwise,} \end{cases} \tag{5}$$

where $b(C_i)$ is equal to (index j may also be used)

$$b(C_i, k) = g_k \sqrt{1/(2Q|C_i|) \ln(|C_{|C_i|}|/\delta)}, \tag{6}$$

where δ is the probability error, $L(\cdot)$ is the label of the segment (0 = water, 1 = vegetation, and 2 = unknown), and $C_{|C_i|}$ is the set of segments with $|C_i|$ pixels. The parameter Q indicates a user-controlled parameter to guide the level of segmentation merging. Since the contextual information already gives a prediction of the object types (water, vegetation, or unknown), we guide the segment merging with a small value for Q, which imposes a strong merging trend on the segmentation results. After merging, the labeled segments are finally grouped into individual semantic regions with a particular regional motion.

3.3 *Context Model: Object-Centric Analysis*

The semantic regions are then used in the following two steps to form the context model in three aspects.

3.3.1 *Step 1: Semantic and spatial context extraction*

As discussed in the Introduction, ships are supposed to travel inside the water region in port surveillance videos. This fact leads to the deduction that unknown regions that are surrounded by water or have common borders with it are potentially ships. Meanwhile, ships cannot be fully buried inside vegetation, which exclude the unknown regions only connected with vegetation. These two deductions can be modeled as spatial and semantic context in our approach, which defines the region C as candidate ships C_{cand} if it satisfies the following two criteria:

$$C_{\mathrm{cand}} \cap C_{\mathrm{water}} \neq \varnothing, \quad \mathrm{Boundary}(C_{\mathrm{cand}}) \nsubseteq \mathrm{Boundary}(C_{\mathrm{veg}}), \tag{7}$$

where C_{water} and C_{veg} represent the water and vegetation regions. The boundary pixels of the candidate ship region are enclosed in $\mathrm{Boundary}(C_{\mathrm{cand}})$ and a similar definition applies to $\mathrm{Boundary}(C_{\mathrm{veg}})$.

3.3.2 *Step 2: Scale context extraction*

Limited by the scope of port surveillance and co-occurred objects, the size of candidate ships should fall in a certain interval, which provides scale contextual information in our context model. The pixel size of C_{cand}, denoted by $|C_{\mathrm{cand}}|$, should satisfy

$$600 < |C_{\mathrm{cand}}| < 0.5 \times \sum_{i=1}^{W} |C_{\mathrm{water}}|, \tag{8}$$

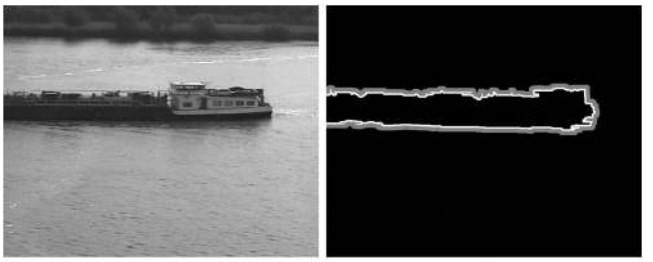
where $|C|$ denotes the number of pixels in the corresponding region and W is the number of segments labeled as C_{water}. The lower bound is used to reduce the effect from small incorrectly classified regions, and the upper bound filters out large incorrect regions that occur, for example, with sunrise. The proposed threshold values were empirically estimated from the available video sequences and fixed for all experiments. We have found that for these sequences, the selected threshold values result in good detection performance over the broad range of different scenarios.

4 Ship Detection Based on Motion Saliency Analysis

In the previous step, we define two dominant regions (water and vegetation) in region labeling. Other objects, such as harbor infrastructures and floating buoys, are classified as unknown regions together with ships and can remain after modeling the contextual information. Therefore, in this section, we will discuss our method that can distinguish moving ships from those objects. As stated previously, one of the criteria for distinguishing ships from other objects adjacent to the water region is that ships generally show more significant motion than the local background. This means that the candidate ship region C_{cand}, whose pixels have salient motion, is detected as a ship. We now describe our approach to salient motion detection.

4.1 *Extraction of Regions of Interest Based on Image Morphology*

Since we have obtained the regions for candidate ships C_{cand} in the previous step, the expensive pixel-wise saliency checking can be avoided by determining motion saliency at the region level. For this purpose, the motion of the candidate ships and the motion of the surrounding background should



(a) (b)

Fig. 6 Results of ROIs extraction: (a) original frame, (b) extracted ROIs with outer part of object in white color and local background in gray color.

be calculated and compared. Therefore, we need to extract the regions of interest (ROIs), which include the outer part of a candidate ship, termed C_{ship}, and the local background around it, called C_{bg}. The reason to consider only the outer part of a ship is that the inner parts of big vessels are often painted in a uniform color, which tends to result in false motion estimation. We use morphological operations to obtain the ROIs, which lead to the expressions

$$C_{\text{ship}} = C_{\text{cand}} - \text{Erosion}(C_{\text{cand}}), \tag{9}$$

$$C_{\text{bg}} = \text{Dilation}(C_{\text{cand}}) - C_{\text{cand}}. \tag{10}$$

In Eqs. (9) and (10), Erosion and Dilation are the corresponding morphological operations. The structuring elements are disks with a radius of 5 pixels for erosion and 10 pixels for dilation. Figure 6 shows the extracted ROIs for a typical frame.

4.2 *Motion Saliency Analysis*

Using the calculated motion values for all pixels, the motion $\mathbf{v}_{\text{ship}}$ of the candidate ship can be defined as the average motion for all pixels inside C_{ship}.

$$\mathbf{v}_{\text{ship}} = \frac{1}{|C_{\text{ship}}|}(\mathbf{v}_1 + \mathbf{v}_2 + \ldots + \mathbf{v}_{|C_{\text{ship}}|}), \tag{11}$$

where $\mathbf{v}_i$ is the motion of pixel i, which belongs to C_{ship}, while $|C_{\text{ship}}|$ represents the number of pixels in C_{ship}. The motion of local background $\mathbf{v}_{\text{bg}}$ is defined similarly.

Another important issue we need to consider is the influence of the dynamic global background, i.e., the motion of water. In certain circumstances, this type of motion can easily lead to difficulties in defining robust criteria for salient motion. To limit the effects of the motion in global background, we can calculate the relative motion in C_{ship} and C_{bg}. Using the extracted context information, we calculate the motion of the whole water region $\mathbf{v}_{\text{water}}$. We denote the relative motion of C_{ship} as $\mathbf{rv}_{\text{ship}}$ and relative motion of C_{bg} as $\mathbf{rv}_{\text{bg}}$.

$$\mathbf{rv}_{\text{ship}} = \mathbf{v}_{\text{ship}} - \mathbf{v}_{\text{water}}, \quad \mathbf{rv}_{\text{bg}} = \mathbf{v}_{\text{bg}} - \mathbf{v}_{\text{water}}. \tag{12}$$

The motion contrast between C_{ship} and C_{bg} is determined as the difference D between the relative motion of C_{ship} and that of its surrounding region C_{bg}.

$$D = |\mathbf{rv}_{\text{ship}} - \mathbf{rv}_{\text{bg}}|. \tag{13}$$

The motion of C_{ship} is salient if the difference D is visually dominant, which leads to the first criterion defining motion saliency.

$$\frac{D}{|\mathbf{rv}_{\text{ship}}|} > T_1. \tag{14}$$

In the first criterion, $\mathbf{rv}_{\text{ship}}$ is employed as the reference motion in the denominator to normalize the motion contrast between the ship and the local background. The normalization is performed for achieving a robust motion saliency definition that is independent of the camera zooming factor in the frame. Threshold T_1 is used to filter nonship objects whose relative motion is not significant (e.g., floating buoys).

We still need to consider two typical cases that can result in false saliency detections: nonship objects (e.g., floating buoys) with small distracting motions in a static water region and static nonship objects (e.g., harbor infrastructure) in the water region with small distracting motions. Therefore, the second criterion for motion saliency is defined as

$$|\mathbf{rv}_{\text{ship}}| - |\mathbf{rv}_{\text{bg}}| > T_2. \tag{15}$$

In our practical conditions, we have empirically set $T_1 = 0.1$ and $T_2 = 0.1$. We have fixed the threshold over the total set of video sequences for different scenarios. The moving ships are detected as the candidate ship regions that satisfy the above two criteria for motion saliency in Eqs. (14) and (15).

4.3 *Presentation Processing: Ship Centroid and Bottom Line Estimation*

In port surveillance scenarios, the centroid of a ship can provide the ship location for radar systems. Moreover, the bottom line estimation has great practical values. We define the bottom of a ship as the lowest row of pixels belonging to the ship body above the water surface. This definition suits a common installation setting in a harbor where the camera acquires images with the ships moving in a horizontal direction. In general, we envisage defining the bottom line of a ship in accordance with its moving vectors. Another property of the bottom line is that its length indicates the length of the ship, which is important for monitoring tasks. Therefore, in our algorithm, the bottom line is estimated only if the ship is fully visible in the frame. We check this condition by verifying whether the bounding box touches any of the image borders. We obtain the corner pixel coordinates of the bounding box by taking the extremities of the contour pixel coordinates of the detected ships. We calculate the centroid of a ship by a simple geometric computation since the coordinates of the corners of its bounding box are given. Visual examples for presentation processing will be shown in the experimental section (Fig. 7).

There is a possibility that the algorithm misses a ship in a frame. It may result in flickering effects in the video because ship detections appear and disappear from frame to frame. The flickering effects deteriorate the visual presentation of our ship detection. We assume that a ship cannot disappear suddenly and cannot move too far between two consecutive

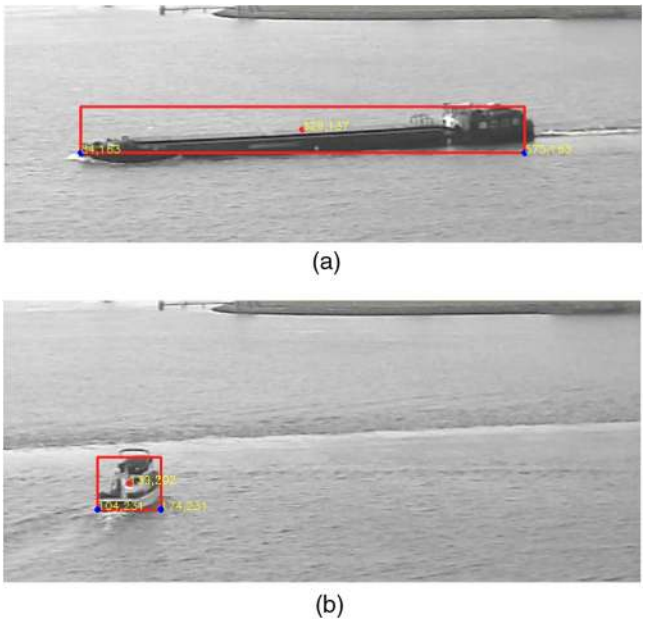


Fig. 7 Presentation processing. Red rectangles represent the region of detected ships; red dots indicate the centroid; blue dots are the left and right ends of the bottom line; yellow numbers indicate the image coordinations, which can be mapped into real world.

frames. These assumptions lead to adding a temporal filter to improve the consistency of our detection results for presentation. Based on another assumption that all ships move in our surveillance video, we remove a detected ship if the centroid of the ship remains in the same position in five consecutive frames, targeting the propagation problem caused by false (positive) detections.

For each detection in the previous frame, we search inside an area that includes the corresponding bounding box in order to determine if the current frame fully or partially contains a ship detection in this area. If there is no detection found and the search area does not touch any of the image borders, a missed detection is spotted, and it is recovered by propagating the bounding box of the previous detection to the current frame.

5 Experimental Results

To evaluate the performance of our ship detection, the algorithm is tested on real-life video sequences recorded in the harbor of Rotterdam, the Netherlands. Since a benchmarking dataset for testing our system does not exist yet, we present the results on our own dataset. All the videos have been captured with a pan-tilt-zoom (PTZ) camera with an standard-definition (SD) resolution of 720×576 pixels and are recorded between 9:00 a.m. and 7:00 p.m. during sunny and cloudy weather without rain, including sunrise and sunset moments. In those videos, the ships are of various types, including container ships, speed boats, tanker ships, fishing boats, and sailing boats, whose distance from the camera is between 0.1 and 1.5 km, with a zoom range of 1 to 35× and with tilting angles in the interval $[-45 \text{ deg}, 0 \text{ deg}]$. The selected video sequences are intervals with fixed camera settings per interval and have various lighting conditions and different backgrounds. All videos are made in the framework of the WATERVisie project, which will be called the WATERVisie dataset in this section. This project aims at building a video-based port surveillance system using a PTZ camera to detect and track ships, supporting the radar system. The total dataset used for evaluation contains 16 video sequences. The total number of sequences is limited

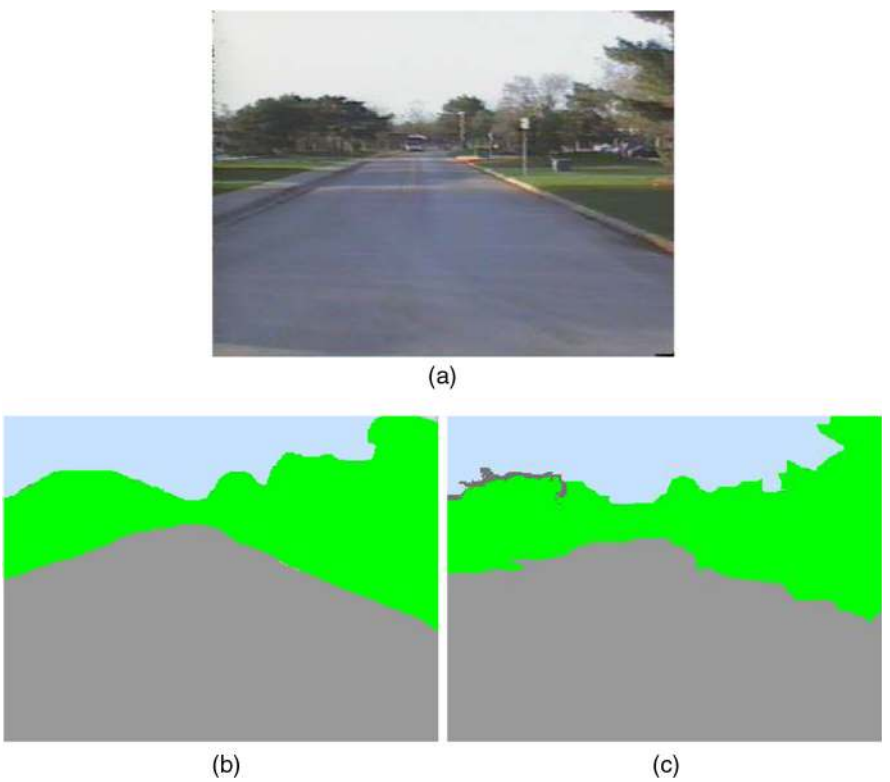


Fig. 8 (a) Image from our dataset. (b) Ground truth of (a). (c) Gravity model-based region labeling. The circled regions show the zoomed imperfections of our region labeling and the same regions in the ground truth.

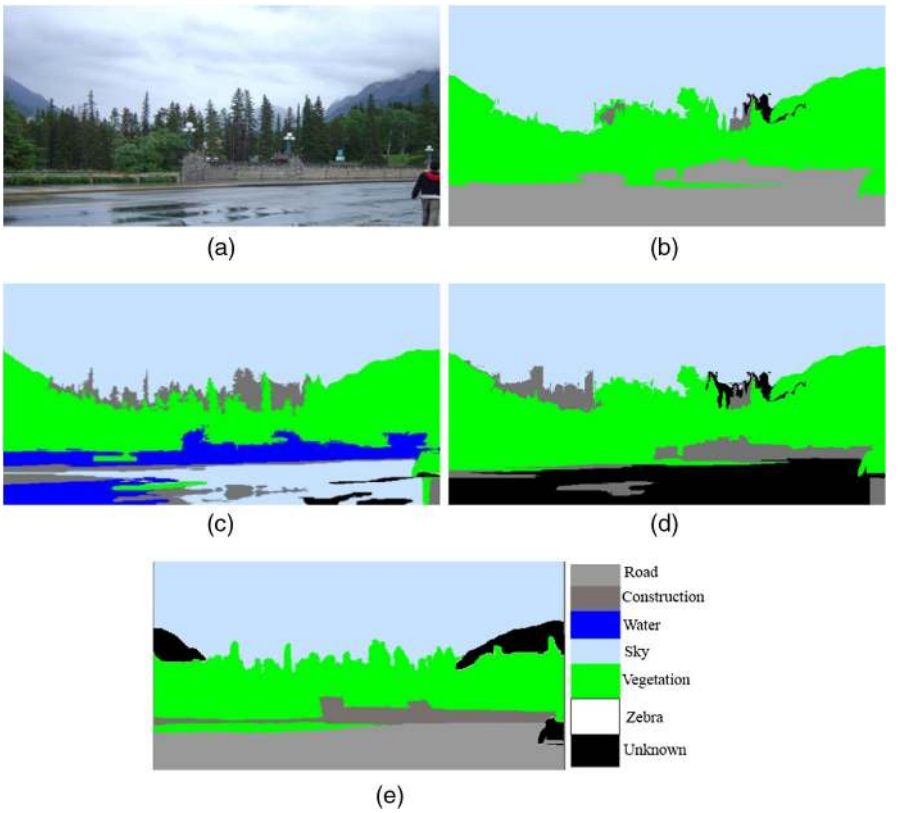


Fig. 9 (a) Image from our dataset. (b) Gravity model-based region labeling. (c) Region labeling from Bao.

since we have restricted access to the system in the harbor. However, the sequences we use contain a significant amount of visual variation, which we believe captures most variation that appears at this physical camera location.

We target a system that can operate on moving cameras. Note that the experimental results are generated on video from a (temporarily static) camera only. We would like to remark that all the proposed algorithmic subsystems are able to operate on video from a moving camera, assuming that the produced images are of sufficient quality and taking the global direction of motion of the camera into account.[43]

5.1 *Object-Centric Region Labeling*

We start with our region labeling experiments. We distinguish between two cases in our evaluation: (1) a generic six-class classification that can be applied in a more broad application range outside the domain of port surveillance and (2) a three-class classification targeting port surveillance. Basically, we are interested in the binary water versus non-water classification. However, since we want to exploit the availability of vegetation in the scene for our motion saliency algorithm, we extend the two-class problem to a three-class problem.

We first discuss the generic application of region labeling. We have constructed a broad dataset that consists of images from multiple Internet datasets and a personal archive. It contains six classes (sky, vegetation, road, water, construction, and zebra-crossing) plus one class (unknown), using dataset of 255 images which contains 121 images for training and 134 images for testing. For the segmentation, there are three parameters to be defined: the standard deviation σ of the Gaussian filter, threshold κ, and the minimum region size. When the minimum region size is small, there will be more regions after segmentation, and it will bring an extra burden for the following classification process. For real-time applications, $\sigma = 1.4$ and $\kappa = 2$, the minimum region size $= 800$ for the WATERVisie dataset and the minimum region size $=$ 300 for the dataset that we used to evaluate our region labeling algorithm. Those parameter settings are a good choice for complex images. The means and variances in

Table 1 Coverability rates of the gravity model in three different color spaces.

Region	HSV (%)	CIE L*U*V* (%)	RGB (%)
Sky	97	94	93
Construction	90	82	89
Water	96	89	92
Road	94	89	90
Vegetation	89	88	86
Zebra	99	99	98
Unknown	98	97	95
Average	95	89	92

the statistics-based model are calculated based on 20 images from the training set.

We now present the experimental results on region labeling. Figure 8 shows results for the generic application with six-class classification. Figure 9 shows results for the three-class classification, specifically targeting port surveillance. Examples of labeling results for two different images are shown in this section. Figure 8(a) shows an original image of our dataset. Figure 8(b) visualizes the ground truth for region labeling of that image, which is achieved by manually segmenting the original image. Figure 8(c) shows the result of the gravity model. Figure 9 illustrates a challenging image along with the results of three different region labeling approaches to highlight the differences between the labeling algorithms. This image is challenging for labeling and contains several ROIs while the color information is quite poor with only small color differences between neighboring regions. It can be observed that our gravity model achieves better results while considerably improving the false detection/rejection rates.

Table 2 Coverability rates for three region labeling approaches.

Region	Gravity model (%)	Statistics-based model (%)	Bao et al.[35] (%)
Sky	97	91	90
Construction	90	88	87
Water	96	93	84
Road	94	95	94
Vegetation	89	84	84
Zebra	99	98	98
Unknown	98	87	94
Average	95	91	90

Note: The gravity model and statistics-based model use HSV color space; Bao *et al.*[35] use RGB color space.

To evaluate the performance of the region labeling algorithm, we use the coverability rate (CR), which measures how much of the true region is detected by the algorithm. This rate is computed by $\mathrm{CR}(O,\mathrm{GT}) = |O \cap \mathrm{GT}|/|\mathrm{GT}|$, where we use the manually annotated ground-truth area (GT), and O is the automatically detected area.[44] In order to analyze the performance of our region labeling algorithm, we have compared our results with the method of Bao et al.[35] We train and test our gravity model on different color spaces on our dataset. Table 1 illustrates the experimental results of the classification approach based on the gravity model on 30 images of the dataset in three different color spaces: CIE $L^*u^*v^*$ (proposed in Ref. 45 as the most efficient color space), RGB, and HSV. Table 1 indicates that compared to the gravity model in RGB color space, the gravity model in HSV color space improves the results by 3%. Therefore, we choose the HSV color space for our classification. Table 2 shows the results of applying the gravity model and statistics-based model approaches compared to Bao's algorithm for our dataset. We can observe that the gravity model results in a higher CR. In the literature, it has been shown that the unitary-category classification of Bao[35] demonstrates a better performance compared to another state-of-the-art approach.[46] We have extended the unitary-category classification of Bao et al.[35] into multicategory classification and applied contextual information as a special feature. With our contribution, we have shown that our gravity model for region labeling outperforms the results achieved by Bao et al.[35]

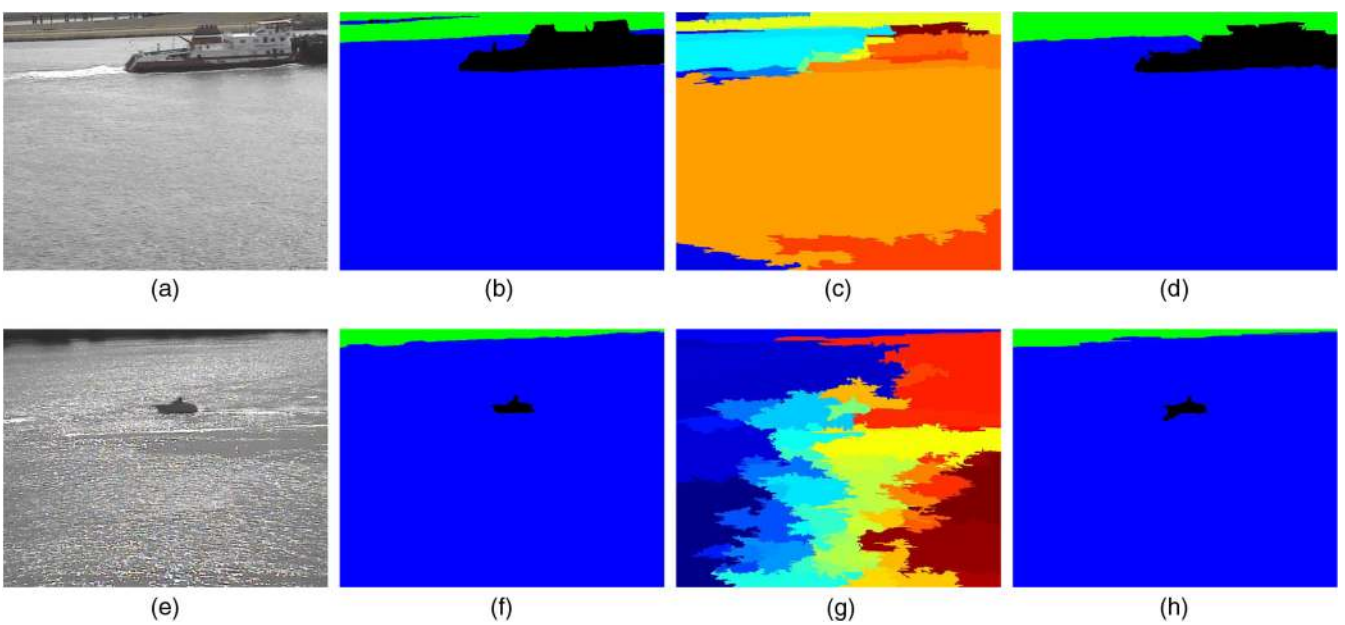


Fig. 10 Region labeling results for WATERVisie dataset. From left column to right column: frames from WATERVisie dataset, ground truth of corresponding frames, intermediate results of graph-based segmentation. The colors are randomly chosen and are not related to semantic class labels. Region labeling results using gravity model (green = vegetation, blue = water, and black = unknown).

We have evaluated the performance of our gravity model on the WATERVisie dataset. The region labeling classifier is trained on 111 typical frames selected from four sequences and tested on 16 videos in this dataset. Those typical frames are representative of various scenarios: water region with/without passing ships, with smooth/rippled surface, and with low/high reflections. Figure 10 visualizes the result of the gravity model on frames from WATERVisie video sequences. This dataset includes three different categories such as water, vegetation, and possible ships, which are labeled as unknown in our approach. Despite the lack of color information, it is obvious that the gravity model performs in a promising manner. Table 3 illustrates the coverability rates for the gravity model on 120 frames of the WATERVisie dataset. The results show that the region labeling is suitable for context modeling in our ship detection system.

Table 3 Coverability rates for gravity model on WATERVisie dataset.

Region	Gravity model on WATERVisie (%)
Water	97
Vegetation	92
Unknown	98
Average	96

Table 4 Ship detection results of Cabin detector.

Test videos	TP + FN	TP + FP	TP	Precision (%)	Recall (%)
$S1$	189	163	152	93.3	80.4
$S2$	455	207	190	91.8	41.8
$S3$	1593	2135	1389	65.1	87.2

Note: TP + FN = manually marked ships, TP + FP = detected ships, TP = correctly detected ships.

5.2 *Ship Detection*

The ship detection test set consists of 16 different video sequences. Those sequences contain a significant amount of visual variation and are categorized into three scenarios: single/multiple ship without occlusion $(S1)$; ships present with occlusions between different ships and/or clutter caused by vegetation $(S2)$; ships during sunrise or sunset moment (highly flickering water) $(S3)$. Since the ship detection system in our project is based on a PTZ camera, it is hard to compare it with other existing systems, which are mainly based on a static camera or untethered camera. Furthermore, there is no benchmark dataset to evaluate the ship detection systems. Therefore, the performances of different ship detection techniques are difficult to compare. To analyze our improved approach, we compare it with our previous algorithms Existing[20] and Cabin detector.[14,19]

5.2.1 *Cabin detector*

The Cabin detector[14,19] employs framework A (Fig. 2) to perform context modeling and ship detection in parallel, whose outputs are combined in a verification process. For the ship detection, the appearance model is constructed using HOG features. First, the image is divided into cells and a gradient orientation histogram is computed for each cell. Each histogram is then normalized to be invariant to contrast changes. When the training images of cabins and background are converted to HOG descriptions, the Cabin detector can be trained using SVM. Based on the trained classifier, cabin detection is performed by sliding a detection window over the image at several scales to locate the cabins independently from the cabin size. For the context modeling, water region is extracted by combining a graph-based segmentation with a

Table 5 Ship detection results of proposed method and Cabin detector.

Test videos	Methods	TP + FN	TP + FP	TP	Precision (%)	Recall (%)
$S1$	a. Improved method	1593	1496	1413	94.5	88.7
	b. Existing	1593	1491	1374	92.1	86.3
	c. Cabin detector	1593	2135	1389	65.1	87.2
$S2$	a. Improved method	455	433	422	97.5	92.7
	b. Existing	455	325	320	98.5	70.3
	c. Cabin detector	455	207	190	91.8	41.8
$S3$	a. Improved method	173	135	130	96.3	75.0
	b. Existing	173	130	122	93.8	71.8
	c. Cabin detector	173	115	97	84.3	56.1

region-based classification. A verification process is designed to check the percentage of water inside the detected cabin regions, where the extracted water region can provide contextual information to reduce the false alarms.

Table 4 shows the detection results for the Cabin detector tested in each video sequence. These test results include the number of manually marked ships, which is the sum of true positive and false negative (TP + FN); the total number of detected ships, which is the sum of true positive and false positive (TP + FP); the number of correctly detected ships, which is the true positive (TP); the precision, and recall. Note that the numbers indicating ships refer to ships presenting in images and not different physical ships.

The numerical results will be analyzed in the following section with a complete comparison among three algorithms. Furthermore, visual results will also be presented and discussed.

5.2.2 *Our improved approach*

Table 5 shows the detection results for our improved approach described in Secs. 2, 3, and 4, and the Existing technique from Ref. 20 and Cabin detector[14,19] tested in each video sequence. Existing is the algorithm that detects water regions as contextual information and applies motion saliency detection as we present in this paper. This approach is a basis of the advanced and improved algorithm presented and tested in this paper. In this evaluation, we only consider the miss or hit, which means that the detection is successful even if the detected ship contains a certain portion of nonship objects.

In the experiment, we manually identify ships on a total set of 2731 frames from all test videos. In these sequences, a single frame may contain a single ship, multiple ships, or no ship. In scenario 1, our ship detection approach successfully detects 1413 ships out of 1593 ships in frames, with a total

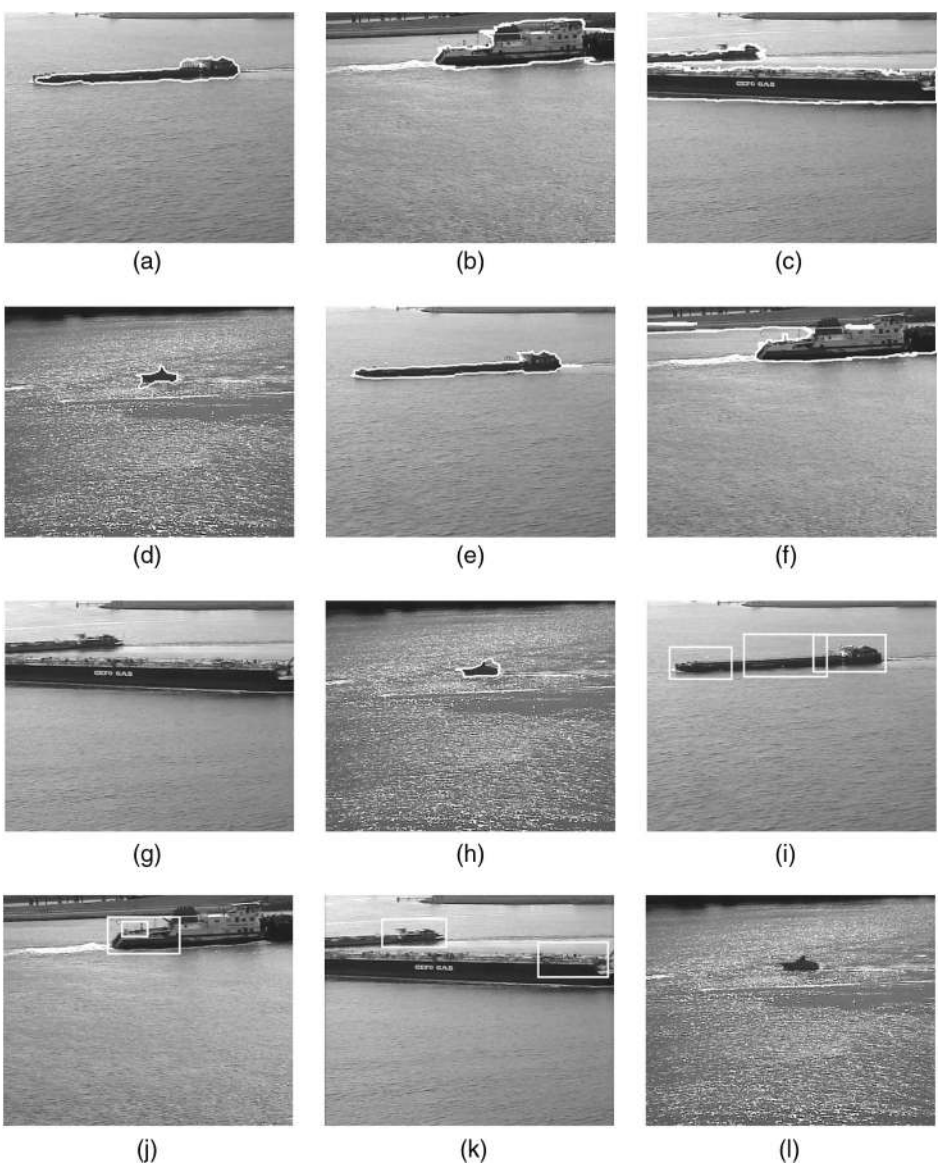


Fig. 11 Visual comparison among our ship detection, Existing method, and Cabin detector. The first row shows the results for our ship detection approach; the second and third rows are the corresponding results of the Existing method and Cabin detector. The typical frames demonstrate the categorized three scenarios from left to right: a long vessel without occlusion (*S*1); a ship cluttered by vegetation (*S*2); two ships occlude each other (*S*2); a sailing ship during sunrise moment (*S*3).

precision of 94.5% and a recall of 88.7%. It gains ~2% in both precision and recall over the Existing method, benefiting from the more advanced context model. For Cabin detector, it obtains similar recall value at the cost of a low precision value. The reason is that the appearance model it builds is simplified but not distinctive from other textured objects. Therefore it tends to generate false detections in vegetation or redundant detections along long vessels. In scenario 3, the numerical results show that our improved approach outperforms the Cabin detector when highly flickering background affects the ship appearance severely [e.g., sunrise in Figs. 11(b) and 11(j)]. Though our algorithm avoids using the detector, which is trained for finding ship appearances (Cabin detector), it still performs well when the target ship differs from the training samples. However, the Cabin detector relies on features in a single frame, where the performance badly deteriorates when affected by the water flickering. Comparing between our improved approach and the Existing method, the higher values in both precision and recall again demonstrate the advantage of a complete context model.

To evaluate the performance of three approaches in scenario 2, Table 6 shows an additional comparison between three approaches in cases of occlusions/clutters. Different from the metrics in measuring precision and recall, only when the occlusions/clutters are solved can the detection distinguish ship from other ship or from nonship objects. The advantage of a complete context model in our improved approach gives a better understanding of context compared to the extraction of water region, which is employed in the other two methods. When there is significant motion in vegetation, the water-ship method will recognize the vegetation region as ships. Moreover, when a ship is cluttered by vegetation, it easily leads to the miss detection or erroneously detects vegetation as part of the ship. As for cabin detection, the clutters affect the ship appearances, which brings difficulties in detecting ships. However, since the vegetation is already labeled as nonship through our context modeling stage, the false detection is avoided before performing the motion analysis. For the occlusion between ships, the motion similarity analysis in our new approach can distinguish different ships while other approaches may fail to work.

We have also made a visual comparison between three approaches as shown in Fig. 11. For all the typical frames, our improved approach can successfully find the whole ship with a bounding box, indicating the delineation of the ship body. However, the Cabin detector can only mark the cabin parts of the ship [Fig. 11(j)] or generate several detection windows along the ship body [Fig. 11(i)]. For small ships moving in a highly flickering water region, the Cabin detector misses the target [Fig. 11(l)], while our improved approach can still find the ship with a boundary indication [Fig. 11(d)]. As for the Existing method, Fig. 11(f) gives an example when a ship is cluttered by vegetation; the approach detects the ship and vegetation as one object because the receiver operating characteristic (ROC) extraction only considers spatial adjacency of nonwater segments. In Fig. 11(g), the detection fails to work because the two ships traveling in two opposite directions are regarded as one ship, which makes the motion of the object not salient compared to the surroundings.

Table 6 Comparison of three approaches in occlusions/clutters cases.

Methods	Occlusions/solved	Clutters/solved
Improved method	203/184	175/168
Existing	203/21	175/5
Cabin detector	203/180	175/14

For the presentation processing, we also show some examples as in Fig. 7 to indicate the possible usage in real-life applications.

All the algorithmic components in our proposed detection system use thresholds. For all thresholds that we use in our system, we have investigated different values. For zooming factors in a given interval, we have fixed each threshold after empirical estimation of the proper value and have found that over the total set of video sequences, detection performance is good. Note that the set of video sequences that we consider covers a large range of possible outdoor scenarios.

6 Conclusion and Future Work

In this paper, we have presented an automatic ship detection system for camera-based port surveillance, featuring the analysis of context information for improving the reliability. The information processing is mapped onto a sequential processing architecture, where the derived context information feeds the subsequent ship detection with specific information about the typical ship locations and candidates in the image, thereby simplifying the ship detection and making it more robust and suitable for real-time implementation. The proposed algorithms are not limited to static cameras and enable the use of moving cameras.

The context information is based on earlier research results and is devoted to extracting various features for an advanced region labeling method. Besides both color and texture, the region labeling also exploits the vertical position as part of a gravity model and a novel statistics model, which involves statistics such as means and standard deviations of the vertical region positions. For the context analysis, a graph-based segmentation is carried out as a preprocessing step to increase the accuracy, by grouping the features with similar local features (color and texture). Meanwhile, the classification at the region level also decreases the computation complexity to enable real-time operation. We have selected the HSV color feature and combined this with Gabor-transformed textures for standard features. For fast region classification, we have applied a random sampling for each segment, and the subsequent multiple-SVM classification is based on a probability model of the segment to be classified as a specific region type. Actually, we model the region-level appearance for each segment, since each region varies considerably depending on environmental and weather conditions. We have found that even in a single image, e.g., pixels representing water tend to have large distances in the feature space, which leads to a low performance of an off-line trained SVM. However, grouping of such pixels belonging to a specific region can have a more stable appearance when fusing multiple features.

One of the key features is also the application of a gravity model to the harbor scenes. Segments classified by a gravity

model are merged into semantic regions based on motion similarity analysis. These regions then provide additional information for finding candidate ships in each frame, which is based on an object-centric context model. The value of this model is that it provides possible locations of candidate ships by exploring the semantic, spatial, and scale constraints between the labeled regions. The unknown segments are merged into candidate ship regions, which have statistically similar motion. Knowing the candidate ships in the region and the contextual ship area, motion saliency detection is the core function in our ship detection. The motion saliency is defined with two criteria that remove nonship objects with small relative motion and static nonship objects surrounded by small distracting motion.

In our system, motion features are important and well explored at three levels: (1) pixel-based motion vectors are computed as basic motion features; (2) segment-level motion is analyzed to group labeled segments into semantic regions based on motion similarity; (3) region-level motion is explored to distinguish ships from other unknown objects based on motion saliency. Because of the detailed motion analysis, we have established a higher robustness for occlusions as we are able to reason about the different moving objects.

The main advantages of our ship detection system are that (1) it requires no prior knowledge of ship appearances and yet it works successfully for various types of moving ships (container ships, speed boats, tanker ships, fishing boats, and sailing boats); (2) it detects the entire ship instead of only a part of the ship (bow, cabin, stern); (3) it produces a full pixel-true segmentation between the ships and their surroundings with a corresponding bounding box and indication of centroid and bottom line; (4) it is able to handle occlusions between different ships and is robust to clutter caused by vegetation; (5) all the proposed algorithmic subsystems are designed to operate on videos obtained by moving cameras. The system is compared to two recent ship detection algorithms and shows robustness and good accuracy for real-life surveillance videos.

Currently, our improved approach cannot handle the detection of temporarily static ships or long vessels across the whole frame with little visual changes. To deal with the limitation, we aim at developing a ship tracking algorithm for combined detection-tracking strategies to further improve consistency and robustness.

References

1. L. Ma et al., "Ship detection by salient convex boundaries," in *3rd Int. Congress on Image and Signal Processing*, Vol. 1, pp. 202–205, IEEE, Piscataway, New Jersey (2010).
2. M. U. Selvi and S. S. Kumar, "A novel approach for ship recognition using shape and texture," *Int. J. Adv. Inf. Technol.* **1**(5), 23–29 (2011).
3. C. Corbane et al., "Fully automated procedure for ship detection using optical satellite imagery," *Proc. SPIE* **7150**, 71500R (2008).
4. M. Bruno et al., "Concurrent use of satellite imaging and passive acoustics for maritime domain awareness," in *Int. Waterside Security Conference*, pp. 1–8, IEEE, Piscataway, New Jersey (2010).
5. M. Teutsch and W. Kruger, "Classification of small boats in infrared images for maritime surveillance," in *International Waterside Security Conference*, pp. 1–7, IEEE, Piscataway, New Jersey (2010).
6. C. Sanderson, D. Gibbins, and S. Searle, "On statistical approaches to target silhouette classification in difficult conditions," *Digit. Signal Process.* **18**(3), 375–390 (2008).
7. H. Li and X. Wang, "Automatic recognition of ship types from infrared images using support vector machines," in *Int. Conf. on Computer Science and Software Engineering*, Vol. 6, pp. 483–486, IEEE, Piscataway, New Jersey (2008).
8. S. Fefilatyev, D. Goldgof, and C. Lembke, "Tracking ships from fast moving camera through image registration," in *Proc. of the 2010 20th Int. Conf. on Pattern Recognition*, pp. 3500–3503, IEEE, Piscataway, New Jersey (2010).
9. H. Wei et al., "Automated intelligent video surveillance system for ships," *Proc. SPIE* **7306**, 73061N (2009).
10. W. Kruger and Z. Orlov, "Robust layer-based boat detection and multi-target-tracking in maritime environments," in *Int. Waterside Security Conference*, pp. 1–7, IEEE, Piscataway, New Jersey (2010).
11. D. Socek et al., "A hybrid color-based foreground object detection method for automated marine surveillance," *Lec. Notes Comput. Sci.* **3708**, 340–347 (2005).
12. N. Arshad, K. Moon, and J. Kim, "Multiple ship detection and tracking using background registration and morphological operations," *Commun. Comput. Inf. Sci.*, Vol. 123, 121–126 (2010).
13. M. D. R. Sullivan and M. Shah, "Visual surveillance in maritime port facilities," *Proc. SPIE* **6978**, 697811 (2008).
14. R. Wijnhoven et al., "Online learning for ship detection in maritime surveillance," in *Proc. of the 32nd WIC Symp. on Information Theory in the Benelux*, pp. 73–80 (2010).
15. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Vol. 1, pp. 886–893, IEEE, Piscataway, New Jersey (2005).
16. O. Marques, E. Barenholtz, and V. Charvillat, "Context modeling in computer vision: techniques, implications, and applications," *Multimed. Tools Appl.* **51**(1), 303–339 (2011).
17. N. Rasiwasia and N. Vasconcelos, "Holistic context models for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(5), 902–917 (2012).
18. J. Yuan, J. Li, and B. Zhang, "Exploiting spatial context constraints for automatic image region annotation," in *Proc. of the 15th Int. Conf. on Multimedia*, pp. 595–604, ACM, New York (2007).
19. X. Bao et al., "Water region and multiple ship detection for port surveillance," in *Proc. the 33rd WIC Symp. on Information Theory in the Benelux*, pp. 20–27 (2012).
20. X. Bao et al., "Ship detection in port surveillance based on context and motion saliency analysis," *Proc. SPIE* **8663**, 86630D (2013).
21. X. Cao et al., "Automatic geo-registration for port surveillance," *Int. J. Pattern Recognit. Artif. Intell.* **24**(04), 531–555 (2010).
22. M. Seibert et al., "SeeCoast port surveillance," *Proc. SPIE* **6204**, 62040B (2006).
23. C. Galleguillos and C. Belongi, "Context based object categorization: a critical survey," *Comput. Vis. Image Underst.* **114**(6), 712–722 (2010).
24. C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.* **20**(3), 273–297 (1995).
25. P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.* **59**(2), 167–181 (2004).
26. R. Nock and F. Nielsen, "Statistical region merging," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(11), 1452–1458 (2004).
27. A. Oliva and A. Torralba, "Building the gist of a scene: the role of global image features in recognition," *Prog. Brain Res.* **155**, 23–36 (2006).
28. L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Vol. 2, pp. 524–531, IEEE, Piscataway, New Jersey (2005).
29. G. Csurka et al., "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision*, pp. 1–22, Springer-Verlag GmbH, Berlin, Germany (2004).
30. G. Heitz and D. Koller, "Learning spatial context: using stuff to find things," *Lec. Notes Comput. Sci.* **5302**, 30–43 (2008).
31. A. Rabinovich et al., "Objects in context," in *IEEE 11th Int. Conf. on Computer Vision*, pp. 1–8, IEEE, Piscataway, New Jersey (2007).
32. L. Wolf and S. Bileschi, "A critical view of context," *Int. J. Comput. Vis.* **69**(2), 251–261 (2006).
33. H.-Y. Cheng and D.-W. Wu, "Region segmentation and labeling in aerial surveillance applications," in *12th Int. Conf. on ITS*, pp. 502–505, IEEE, Piscataway, New Jersey (2012).
34. N. Nguyen and Y. Guo, "Comparisons of sequence labeling algorithms and extensions," in *Proc. of the 24th Int. Conf. on Machine Learning*, pp. 681–688, ACM, New York (2007).
35. X. Bao et al., "Water region supporting ship identification in port surveillance," in *Advanced Concepts for Intelligent Vision Systems*, pp. 444–454, Springer-Verlag GmbH, Berlin, Germany (2012).
36. J. R. Smith and S. Chang, "Single color extraction and image query," in *Int. Conf. on Image Processing*, Vol. 3, pp. 528–531, IEEE, Piscataway, New Jersey (1995).
37. G. Rellier et al., "Texture feature analysis using a Gauss-Markov model in hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.* **42**(7), 1543–1551 (2004).
38. W. K. Kong, D. Zhang, and W. Li, "Palmprint feature extraction using 2-d Gabor filters," *J. Pattern Recognit.* **36**(10), 2339–2347 (2003).
39. X. Bao et al., "Robust moving ship detection using context-based motion analysis and occlusion handling," in *6th Int. Conf. on Machine Vision* (2013).

40. C. Liu, "Beyond pixels: exploring new representations and applications for motion analysis," PhD Thesis Massachusetts Institute of Technology (2009).
41. T. Brox et al., "High accuracy optical flow estimation based on a theory for warping," *Lec. Notes Comput. Sci.* **3024**, 25–36 (2004).
42. A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods," *Int. J. Comput. Vis.* **61**(3), 211–231 (2005).
43. L. Klicnar, "Robust detection of moving object in video," Master's Thesis, Brno University of Technology, Czech Republic (2012).
44. S. Javanbakhti, S. Zinger, and P. H. N. de With, "Fast sky and road detection for video context analysis," in *Proc. of the 33rd WIC Symp. on Information Theory in the Benelux*, pp. 210–218 (2012).
45. C. Benedek and T. Sziranyi, "Study on color space selection for detecting cast shadows in video surveillance," *Int. J. Imaging Syst. Technol.* **17**(3), 190–201 (2007).
46. A. L. Rankin, L. H. Matthies, and A. Huertas, "Daytime water detection by fusing multiple cues for autonomous off-road navigation," in *Proc. of the 24th Army Science Conf.*, pp. 177–184 (2004).

Xinfeng Bao received his BSc degree in electronics and information engineering from Huazhong University of Science & Technology, China, in 2008. In the same year, he started his master's study in computer engineering at Politecnico di Milano, supported by ICE-UNIONCAMERE scholarship. In 2010, he joined Graphics 4 Embedded Systems research group at STMicroelectronics in Italy as an intern. In early 2011, he received his MSc degree from Politecnico di Milano. Since 2011, he has been a PhD student in the Video Coding and Architectures (VCA) group at the Eindhoven University of Technology (TU/e), the Netherlands. He is currently working on maritime/road traffic analysis and surveillance.

Solmaz Javanbakhti received her MSc degree in computer engineering from Shahid Beheshti University, Tehran, Iran, in 2010. She did her master's thesis on segmentation and quantifying of MR images at the Biomedical Engineering Department in TU/e, Netherlands. Since 2010, she has been a PhD student in VCA group at the Department of Electrical Engineering in TU/e. She is currently working on video surveillance applications including video context analyzing and region labeling for semantic understanding of the video.

Svitlana Zinger received her MSc degree in computer science in 2000 from the radiophysics faculty of the Dnepropetrovsk State University, Ukraine. She received her PhD degree in 2004 from the Ecole Nationale Superieure des Telecommunications, France. Her PhD thesis was on interpolation and resampling of 3-D data. In 2005, she was a postdoctoral fellow in the Multimedia and Multilingual Knowledge Engineering Laboratory of the French Atomic Agency, France, where she worked on creation of a large-scale image ontology for content-based image retrieval. From 2006 to 2008, she was a postdoctoral researcher at the Center for Language and Cognition Groningen and an associated researcher at the artificial intelligence department in the University of Groningen, the Netherlands, working on information retrieval from handwritten documents. She is currently an assistant professor at the VAC research group in TU/e.

Rob Wijnhoven graduated in electrical engineering from TU/e in 2004. From 2004 to 2009, he worked on object categorization for video surveillance at Bosch Security Systems, Eindhoven, The Netherlands. In 2009, he joined ViNotion, Eindhoven, and is working on object detection in various application fields. Since 2004, he has been active in several related international projects. His interests include pattern recognition and machine learning for computer vision applications, and he is currently working toward a PhD degree.

Peter H. N. de With graduated in Electrical Engineering (MSc., ir.) from Eindhoven University of Technology and received his Ph.D. degree from University of Technology Delft, The Netherlands. From 1984 to 1997 he worked for Philips Research Eindhoven, where worked on video compression chaired a cluster for programmable TV architectures as senior TV Systems Architect. From 1997 to 2000, he was full professor at the University of Mannheim, Germany, Computer Engineering, and chair of Digital Circuitry and Simulation. From 2000 to 2007, he was with LogicaCMG in Eindhoven as a principal consultant Technical SW and distinguished business consultant. He was also parttime professor at the University of Technology Eindhoven, heading the chair on Video Coding and Architectures. In the period 2008–2010, he was VP Video (Analysis) Technology at CycloMedia Technology. Since 2011, he is assigned full professor at Eindhoven University of Technology and appointed scientific director Care & Cure Technology and theme leader Smart Diagnosis in the University Health program. de With is a national and international expert in video surveillance for safety and security and has been involved in multiple EU projects on video analysis, featuring object and behavior recognition, and also surveillance projects with the Harbor of Rotterdam, Dutch Defense, Bosch Security Systems, TKH-Security, etc. He is board member of DITSS and R&D advisor to multiple companies. De With is Fellow of the IEEE, has (co-) authored over 300 papers on video coding, analysis, architectures, 3D processing and their realization. He is the (co-)recipient of multiple papers awards of the IEEE CES, VCIP and Transactions papers.