Ocean Engineering 54 (2012) 1–12

Contents lists available at SciVerse ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system

Sergiy Fefilatyev [a,*], Dmitry Goldgof [a], Matthew Shreve [a], Chad Lembke [b,1]

[a] Department of Computer Science and Engineering, University of South Florida, 4202 E.Fowler Ave., ENB 118, Tampa, FL 33617, USA
[b] Center for Ocean Technology, University of South Florida, 140 7th Avenue South St. Petersburg, FL 33701, USA

ARTICLE INFO

Article history:
Received 25 November 2011
Accepted 30 June 2012
Editor-in-Chief: A.I. Incecik
Available online 24 July 2012

Keywords:
Maritime surveillance
Ship detection
Target tracking
Horizon detection
Uncalibrated camera
Moving platform

ABSTRACT

Visual surveillance in the maritime domain has been explored for more than a decade. Although it has produced a number of working systems and resulted in a mature technology, surveillance has been restricted to the port facilities or areas close to the coast line assuming a fixed-camera scenario. This paper presents a novel algorithm for open-sea visual maritime surveillance. We explore a challenging situation with a forward-looking camera mounted on a buoy or other floating platform. The proposed algorithm detects, localizes, and tracks ships in the field of view of the camera. Specifically, developed algorithm is uniquely designed to handle rapidly moving camera. Its performance is robust in the presence of a random relatively-large camera motion. In the context of ship detection we developed a new horizon detection scheme for a complex maritime domain. The performance of our algorithm and its comprising elements is evaluated. Ship detection precision of 88% is achieved on a large dataset collected from a prototype system.

1. Introduction

Surveillance systems play an important role in management and monitoring of littoral and maritime areas by providing tools for situation awareness, threat assessment, and decision making. Various sources of surveillance, monitoring, maritime safety information are available. These include the Automatic Identification System (AIS), Vessel Monitoring Systems (VMS), air- and space-borne SAR systems, ship- and land-based radars, air- and space-borne optical sensors, harbor-based visual surveillance. During the last decade technology has been moving towards integration of several information sources, of which one essential component is vision. Fusion of vision with other information sources allows more accurate and descriptive monitoring of coastal areas, maritime borders, and offshore assets. Examples of this trend include systems that integrate AIS/VMS with SAR-imagery (Lemoine et al., 2005; Saur et al., 2011), radar- and visual-based surveillance for ports (Seibert et al., 2006; Rodriguez Sullivan and Shah, 2008), land-radars with visual information from air-borne platforms, and ship-based systems that integrate visual and other sensor's information (Wei et al., 2009; Liu et al., 2008).

A new concept (Fefilatyev et al., 2009; Kruger and Orlov, 2010) helps to integrate information from radars/AIS with vision sensors placed on autonomous buoys. Such an autonomous unmanned buoy-system is an attractive choice for integrated surveillance for reasons of cost, persistent ocean presence, form-factor, and flexibility. A network of vision sensors with on-board processing and bi-directional communication can provide a real-time intelligence for critical maritime areas. It can detect, classify, identify and track small vessels at sea often associated with issues of illegal immigration, smuggling, and poaching. Examples of use of vision information from such source include verification of vessel's class, intent, and behavior, as well as description of state and status of the vessels. A passive vision sensor is also more desirable in military applications because it does not reveal the location of the surveillance system.

The aim of this paper is to present a novel algorithm for on-board processing of image- and video-data obtained by a buoy-based maritime surveillance system equipped with a low-sitting forward-looking camera. The system operates autonomously and intelligently in a maritime area away from the visible coast-line. This involves detection, localization, and tracking of ships or other marine vehicles of interest in imagery taken by its color camera that is mounted on a buoy and is a subject to rapid random motion associated with the buoy's flotation. The output of the algorithm are images of the found unique target candidates. The algorithm, described in the paper, addresses the most common problems in buoy-based visual surveillance: horizon detection in complex maritime scenes; frame stabilization in video with large

* Corresponding author. Tel.: +1 813 974 3652; fax: +1 813 974 5456.
E-mail addresses: sfefilatyev@gmail.com (S. Fefilatyev), goldgof@cse.usf.edu (D. Goldgof), mshreve@cse.usf.edu (M. Shreve), clembke@marine.usf.edu (C. Lembke).
[1] Tel.: +1 727 553 1009; fax: +1 727 553 3967.

0029-8018/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.oceaneng.2012.06.028

inter-frame motion; segmentation of low-contrast targets in imagery corrupted by compression artifacts; inter-frame tracking. The algorithm is formulated for real-time execution on a limited-memory commercial of-the-shelf computing platform and is suited for low-powered autonomous systems deployed for long periods of time. The performance of proposed algorithm is evaluated on a large set of annotated image data obtained from a prototype. The results are shown for the most important characteristics: detection, localization, and tracking accuracy.

The rest of the paper is structured as follows. Section 2 relates our work to the existing work on visual maritime surveillance. Section 3 presents the algorithms for horizon detection, frame stabilization, segmentation, detection, and tracking of ships. The description of metrics used to evaluate the performance of the proposed approach is given in Section 4. Section 5 reports experimental evaluations on data obtained from a prototype system. Conclusions are drawn in the last section.

Initial work on this problem was presented in earlier publications (Fefilatyev et al., 2009, 2010).

2. Related work

We review the developments in the area with respect to several aspects. First, in Section 2.1 we review the general topic of visual maritime surveillance with stationary cameras, mostly related to port security. Second, in Section 2.2 we review surveillance from non-stationary platforms: air- and sea-borne. Third, because of the importance of horizon detection in our algorithm, in Section 2.3, we briefly review the existing approaches to horizon line detection.

2.1. Visual surveillance in maritime domain with stationary cameras

The harbor, port, and coast surveillance have been extensively studied for more than a decade (Voles et al., 1999; Socek et al., 2005; Seibert et al., 2006; Feineigle et al., 2007; Rodriguez Sullivan and Shah, 2008; Chen, 2008; Sanderson et al., 2008; Bloisi et al., 2009). In general, all the existing systems for port surveillance assume that the camera is in a fixed position, and the water surface is fixed in terms of a camera view. With this set of assumptions, algorithms for background subtraction or background suppression are mostly used. Many of the mentioned papers describe already existing systems (Seibert et al., 2006; Rodriguez Sullivan and Shah, 2008; Bloisi et al., 2009; Cao et al., 2010), signifying matureness of the topic and spread of the commercial applications. Reported functionality of such systems include vessel detection, discrimination between vessel classes, comparison of the detections with the notifications of arrival received by the port, and geo-registration of ships that navigate in port waters. Those methods rely on stationary camera, the assumption that is not valid in our settings.

2.2. Visual surveillance in maritime domain from non-stationary platforms

Many applications for visual maritime surveillance from non-stationary platforms stem from Automated Target Detection (ATR) systems (Li et al., 2004; Valin et al., 2006; Tao et al., 2007; Li and Wang, 2008; Araghi et al., 2009; Shaik and Iftekharuddin, 2009). ATR systems are designed to be air- or sea-borne. They rely on forward-looking infrared (FLIR) cameras as a primary source of the imagery because the infrared sensors are less-sensitive to variations in illumination and appearance changes, which is common in maritime scenes. A notable disadvantage of FLIR-based systems is the low resolution of images and significant power consumption which makes these systems less capable for autonomous operation. The most significant applications not related to ATR or harbor surveillance are classification of ships (Enríquez de Luna et al., 2005; Gupta et al., 2009), image stabilization (Morris et al., 2007; Cao and Zhang, 2007), ship motion estimation (Benetazzo, 2011), and, relevant to our work, detection and tracking in non-stationary maritime environment (Liu et al., 2008; Kruger and Orlov, 2010; Teutsch and Kruger, 2010; Wei et al., 2009).

The systems described by Liu et al. (2008), Kruger and Orlov (2010), and Teutsch and Kruger (2010) differ from ours in terms of data source. Liu et al. (2008) rely on color visual data from omni-directional camera. Although their approach addresses many segmentation problems and provides a 360° surveillance with a single camera, the authors admit a limited the effective range because of very wide-angle of observation. Increase in the range requires substantial increase in the pixel resolution of the camera and height of the mast where the omni-directional camera is installed. Kruger and Orlov (2010) and Teutsch and Kruger (2010) rely on a narrow-angle buoy-based FLIR camera and share some ideas with our earlier work (Fefilatyev and Goldgof, 2008; Fefilatyev et al., 2009). However, they focus on detection and tracking in data from FLIR cameras which requires different approach to segmentation.

The system described by Wei et al. (2009) has similar assumptions to the ones used in this work. However, their surveillance system is designed for ships, and performance of the described detection and tracking algorithm may be limited in the case of buoy-based mechanically not stabilized camera that experiences high-magnitude rapid motion in pitch, yaw, and bank dimensions. The specific background-subtraction algorithm they proposed relies on the horizon line that is always present in an image, is sensitive to fast illumination changes caused by the dynamics of maritime scenes, and, thus, does not show robustness on the data obtained from a buoy-based camera. Wei et al. (2009) also assume a high-sitting camera with objects of interest located below the horizon line. Our paper aims to cover the issue of a very dynamic scenes with targets on-or-above the horizon line, and provides performance evaluation of the proposed approach.

2.3. Approaches to horizon detection

The two main application domains for horizon estimation are vanishing line estimation and unmanned aerial vehicle (UAV) navigation.

The vanishing line approaches (Lv et al., 2006; Cao et al., 2010; Guo and Chellappa, 2006; Criminisi and Zisserman, 2000; Sheikh et al., 2006) are aimed to locate a vanishing line of the horizon which may or may not be the actual visible horizon. All the methods from this category of approaches are related to a fixed camera and, thus, are not applicable in our case. The only method that could be used in a non-stationary environment is the approach described by Criminisi and Zisserman (2000), which utilizes the assumption of homogeneity of surface's texture. However, such geometry-based solution does not provide sufficient precision for stochastic textures.

In the case of UAV navigation, the horizon line is used as an alternative to inertial sensors (Ettinger et al., 2003; McGee et al., 2005; Bao et al., 2005; Zhang et al., 2010; Dusha et al., 2007). There are several approaches in this category to find the horizon line: by using analysis of projection profiles of the edge map of an image (Bao et al., 2005; Zhang et al., 2010), using Hough transform (Wei et al., 2009) and optical flow (Dusha et al., 2007). Those algorithms are inappropriate for our case because of either high computational demands, or assumption on always present horizon. The horizon detection algorithm described by McGee et al. (2005) is fast and addresses both situations: when the horizon line is visible and when it is absent. The approach classifies the pixels into sky/ground regions and finds the boundary between

them. Since the classification power is incorporated in the beginning of such a scheme, it easily detects non-horizon images. Another representative algorithm, developed by Ettinger et al. (2003), relies on a color-based statistical model of the sky and ground (water in our case). The algorithm searches for the line which maximizes the sky–ground separation criterion. The authors also offer a solution for extreme attitude case—when only sky or ground regions are visible by validating the current results of the horizon detection with the accumulated statistics of the sky/ground appearance.

3. Algorithms

In our application of a buoy-based visual maritime surveillance, the camera is installed approximately parallel to the ocean's surface. Such a forward-looking camera has a limited field of view and has a limited resolution. The acquired data comes from the visible part of the spectrum and is represented by the RGB-color model. The focus of the camera is set to infinity, and thus, allows detection only of far-located objects located just above the horizon line. Because the camera does not have stabilization and is firmly attached to its non-stationary buoy-platform the scenery it observes is directly dependent on the uncontrolled motion of the buoy. In this work we assume that such motion of the camera is erratic. Thus, motion pattern of the buoy provides us with the three categories of possible images. The situation when the optical axis of the moving camera lies close to parallel to the ocean surface will result in horizon images, i.e. images where the horizon line separates the two most significant regions of the image—sky and water (Fig. 1(a)–(d)). The other two categories of images are obtained when only sky region is visible (camera points off the water surface, see Fig. 1(e)) or only water region is visible (see Fig. 1(f)).

An important constraint in our model is that possible objects of interest, i.e. ships, comprise only a small fraction of the field of view, and thus, most of the scenery is represented by a water and sky regions. An exception occurs when such a constraint is not held (see Fig. 1(g), (h)) and such images are not considered for further processing.

We address all the mentioned cases by relying on one important feature that is always present in maritime scenery: the horizon line. Our approach can be summarized as the following:

- Images with high confidence in the location of the horizon line are considered for further analysis.
- High confidence in the absence of the horizon line in an image is an indication that the camera is pointing to the sky or water region only, and thus, no further search of ships is possible and such image is disregarded.
- Low confidence in the location of the horizon line addresses the situation when an exceptional situation occurs: field of view is blocked or quality of imagery is corrupted by a water splash. Such image is disregarded as well.

3.1. Overview of algorithm

The algorithm for the detection and localization of ships uses a four-step strategy to find possible targets. Each of those components may either produce output for the consecutive step or rule out the current image frame, if it is found that the model assumptions are not satisfied. In case when assumptions are found broken, the frame is labeled as 'intractable' and the algorithm moves to the next frame in the image sequence. Fig. 2 shows the basic structure of our approach and the order in which the components of the algorithm are used.

The horizon detection, described in Section 3.2, is the most important step in the algorithm due to several reasons. The found horizon line is used to determine if the current frame satisfies the model assumptions about water and sky regions. The horizon line is also used in the following image registration step as a reference line for image alignment. Finally, the horizon line is used in the segmentation step to reduce the search space for possible targets as all objects of interest are expected to be above the horizon line and be also adjacent to it.

The main purpose of the image registration step is the possibility of subsequent tracking of the detected targets, because the correctly registered frames of the video sequence allow simplified tracking of the targets on the horizon line. For that, all frames of the video sequence are registered in the common coordinate system (Zitova and Flusser, 2003). The other purpose of the registration step is filtering of intractable frames: high confidence in the correct alignment of horizon images is another verification that the observed scenery satisfies model assumptions. Details on the implementation of the image registration module are given in Section 3.3.

The segmentation step of the algorithm, described in Section 3.4, localizes the regions of the image above the horizon that potentially contain ships. Such localization is based on the appearance of ships, affected by illumination, orientation, weather conditions, and is a subject to compression artifacts of

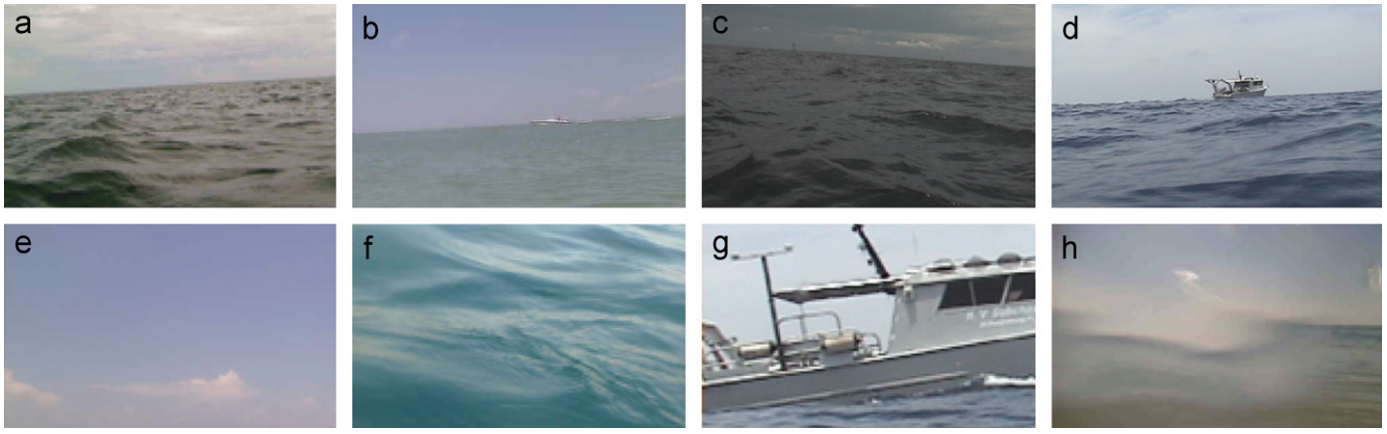

Fig. 1. Camera that is attached to the untethered buoy will provide images of several possible categories of horizon images. (a)–(d) Images with horizon clearly discernable. (e) Camera points to the sky and sky is the only scenery visible in the image. (f) Image of water surface only. (g) Objects such as ships when they take a big fraction of the field of view may seriously affect horizon detection. (h) Blur that comes from water splashes is another factor affecting horizon detection.

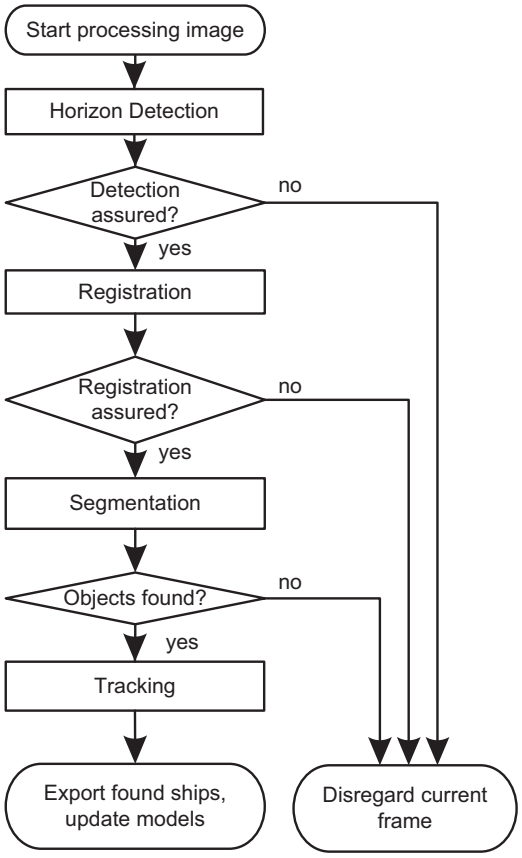

Fig. 2. Outline of the detection and tracking algorithm. Algorithm consists of four main modules: horizon detection, image registration, segmentation, and tracking. Depending on the confidence of horizon detection or registration the current frame can be dropped from consideration.

the video encoder of the camera. The output of the segmentation step is important for subsequent tracking, because the segmented targets are used as an input of the tracking algorithm.

The tracking step, described in Section 3.5, is used to increase the accuracy of ship detection and reduce false alarms. The tracking module accepts input from the segmentation algorithm and rules out targets that do not have a consistent history of location, size, and appearance. The tracking algorithm provides the final output of detected and localized targets in the obtained imagery.

The modules in Fig. 2 provide confidence for their output. This allows to streamline the processing of the data by stopping processing the frames which have been found ill-fit for the assumed model. For example, sometimes in cases of abundance of sun glare in the image near the horizon line the further processing must be halted. This is implemented through checking the confidence of horizon detection: low confidence in the horizon, or high confidence in the absence of horizon indicates assumption violation. Similarly, low confidence in the correct registration of the frame in the global coordinate system is a reason for the frame to be disregarded from further processing because the segmentation and tracking results depend on it.

3.2. Horizon detection

The approach to the horizon detection described in this work combines features from works of McGee et al. (2005), Ettinger et al. (2003), and Wei et al. (2009) and addresses situations mentioned in the beginning of Section 3.

Similar to the work by McGee et al. (2005), the horizon is modeled as a straight line in the normal representation:

$$x\cos\Theta + y\sin\Theta = \rho \tag{1}$$

where (x,y) are the coordinates of the points on the line, Θ is the angle between the perpendicular to the line and horizontal axis of the coordinate system, ρ is the perpendicular distance from the origin to the line. The RGB color is used as a measure of appearance as in paper by Ettinger et al. (2003). For any given hypothesized horizon line, the pixels above the line are labeled as sky, and pixels below the line are labeled as water. All sky pixels in such model are denoted as

$$x_i^s = [r_i^s g_i^s b_i^s]^T, \quad i \in \{1,\ldots,n^s\} \tag{2}$$

where r_i^s denotes the value for intensity of the red channel in RGB, g_i^s denotes the green channel value, and b_i^s denotes the blue channel value of the i-th sky pixel, and n^s is the number of pixels in sky region of the image. All the hypothesized water pixels are noted in a similar manner:

$$x_i^w = [r_i^w g_i^w b_i^w]^T, \quad i \in \{1,\ldots,n^w\} \tag{3}$$

This provides two distributions of pixels, where each pixel is represented by a vector of RGB colors. Both sky and water distributions are assumed to be roughly Gaussian distributed in RGB color space. In our experiments with separation criterions we found the following measure, based on Bhattacharyya (1943) distance, to best maximize difference between the distributions of water and sky:

$$J(\Theta,\rho) = (\mu_s-\mu_w)^T(\Sigma_s+\Sigma_w)^{-1}(\mu_s-\mu_w) \tag{4}$$

where Σ_s and Σ_w are the covariance matrices of the sky and water pixel distributions:

$$\Sigma_s = \frac{1}{n_s-1}\sum_{i=1}^{n_s}(x_i^s-\mu_s)(x_i^s-\mu_s)^T \tag{5}$$

$$\Sigma_w = \frac{1}{n_w-1}\sum_{i=1}^{n_w}(x_i^w-\mu_w)(x_i^w-\mu_w)^T \tag{6}$$

μ_s and μ_w are the vectors for mean color-intensity of the sky and water pixel distributions and are defined as

$$\mu_s = \frac{1}{n_s}\sum_{i=1}^{n_s}x_i^s, \quad \mu_w = \frac{1}{n_w}\sum_{i=1}^{n_w}x_i^w \tag{7}$$

However, instead of searching for horizon line in the line-parameter space to maximize the computationally expensive separation criterion as did Ettinger et al. (2003), we adopted an approach of, first, selecting a limited number or candidate-lines using a less computationally intensive approach, and then, checking them against the separation criterion (4).

This, 'candidates-first', approach provides two benefits. First, as mentioned, the selection of candidates significantly reduces computational load as compared to two-dimensional search in line-parameter space with (4). Second, in most practical cases such approach provides better accuracy of horizon line detection when compared to the pure implementation of method by Ettinger et al. (2003). The reason for better accuracy of horizon localization is explained by the fact that the algorithm of Ettinger et al. (2003) assumes two regions, the water and the sky, which are homogeneous in color. This assumption seldom holds true. Because of perspective effects of forward-looking camera and uneven illumination the water regions vary in color significantly. Similarly, the sky regions are often non-homogeneous in color. This usually results in the horizon line found close to the correct position but shifted far enough that the subsequence segmentation of ships is impossible.

In order to select the candidate lines the Hough transform on the edge map of the image is performed, similar to Wei et al. (2009) and McGee et al. (2005). K largest distinct peaks in the Hough accumulator space, corresponding to the strongest lines in the image, are found sequentially. While searching for the next largest peak, the non-maximum suppression in the neighborhood of the previous peak is applied to exclude selection of very close neighbors of the candidate lines. Once K candidates are identified, the sky–water separation criterion (4) is computed for each candidate and the candidate with the maximum score of (4) is selected for further processing.

In order to addresses the situations when only sky or only water is present in the image we use the approach from paper by Ettinger et al. (2003). The approach relies on running statistical models of the sky and water and a set of criterions for determining the situation of extreme attitude. The initial statistical models of the sky and water are obtained from 20 first images of a video sequence under assumption that the horizon is present.

The exceptional situation with the blurred images or with images where external object is blocking the view is addressed by checking the value of the sky–water separation criterion against the threshold $T1$. When the sky/water regions do not show much distinction, the model of two regions separated by a line does not hold, and thus, no ships can be detected on such horizon. In order to choose the value for $T1$ we created a separate dataset of images where a boat blocks significant part of the field of view or where the camera has been splashed by water. Once trained offline on such dataset, $T1$ was used throughout the rest of the experiments.

The candidate line for horizon with the maximum sky–water separation criterion (4) which passed the tests for extreme attitude and exceptions is declared the horizon. Fig. 3 summarizes all steps of the horizon detection algorithm.

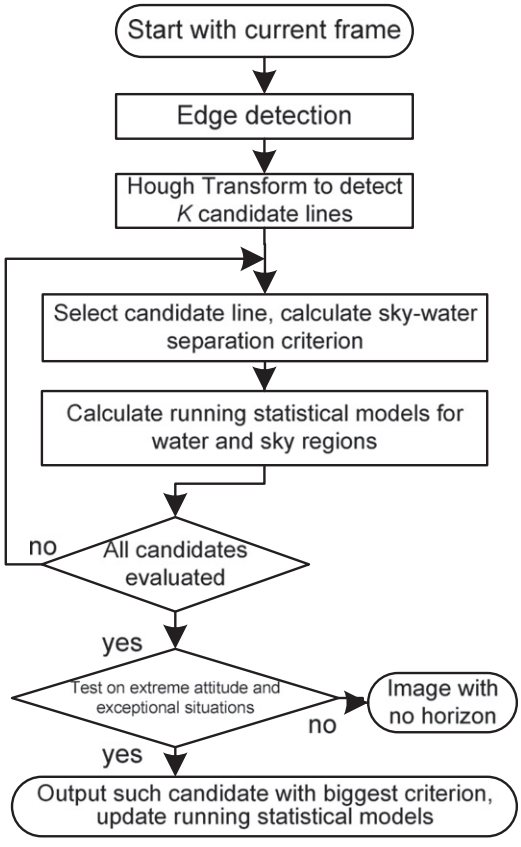

Fig. 3. Outline of the horizon detection algorithm. Several candidates lines for horizon are detected with the Hough transform. Using a sky–water separation criterion the final candidate horizon line is selected. To detect situations when the horizon is not present in the image the final candidate is checked against the running statistical model.

3.3. Image registration

The methods of image registration are divided into two major categories (Zitova and Flusser, 2003): area-based and feature based. In our approach, where all the frames are related to each other through affine transformation, the area based algorithm is sufficient. Use of horizon line allows to simplify the area-based registration significantly. Using the horizon line as the axis in the coordinate system common for all frames reduces the search for alignment into one dimension, along the horizon line (see Fig. 4(a)–(c)). Also, since the horizon line is found prior to the registration step, no additional computations are required. One-dimensional normalized cross correlation is used as a similarity measure between the registered images:

$$\gamma(u) = \frac{\sum_x [f(x)-\bar{f}_u][t(x-u)-\bar{t}]}{\{\sum_x [f(x)-\bar{f}_u]^2 \sum_x [t(x-u)-\bar{t}]^2\}^{0.5}} \quad (8)$$

where f is the base image with the coordinate system fixed on it, t is the second image (template) which is being registered against f, u is the alignment distance, $\bar{t}$ is the mean of the template, $\bar{f}_u$ is the mean of $f(x)$ in the region under template. Fig. 4(e) shows the values of (8) for correlation of Fig. 4(a) and (b). The peak in its value corresponds to the best alignment of images along the horizon. For correlation purposes the images are converted into real-valued gradient images where the intensity of the channel is computed using the magnitude of color gradient described by Di Zenzo (1986). Only rectangular patches of the gradient images which overlay sky region adjacent to horizon are used for registration. The height of the rectangular patches is parameterized by parameter L. The magnitude of the color gradient was selected as a method to emphasize the edges of ships silhouettes in low-contrast maritime environment. However, to reduce computation, the Laplacian filter applied to the grayscale image can also be used. Before taking the gradient the image strip is processed through the Gaussian filter to smooth effects of compression artifacts.

The first two frames of the video initialize the global coordinate system and create the base image. Later frames are used to grow the base image and to update its pixel values. The pixels of the base image are updated using the following formula:

$$B_t(x,y) = \alpha B_{t-1}(x,y) + (1-\alpha)T(x,y) \quad (9)$$

where $B_t(x,y)$ and $B_{t-1}(x,y)$ are the intensity of the pixel at the position (x,y) in the base image at times t and $t-1$, $T(x,y)$ is the intensity of the pixel in the template, and α is the rate of update. This technique makes the registration adaptive to the change of the environment.

The threshold value $T2$ is used to check the value of cross-correlation for significance of registration. Those frames of the video sequence, where the maximum of cross-correlation is below the threshold, are disregarded from further consideration. Fig. 4(e) shows an example where the maximum of the similarity measure between Fig. 4(a) and (b) is above the threshold, confirming the correct horizon detection. The value for $T2$ is learned empirically on a small dataset of maritime images.

Another important use of the threshold $T2$ is for those images where the sky regions above the horizon are completely uniform in appearance. Use of the area-based methods is limited only to the regions of the image above the horizon line which provides some non-uniform texture (for example, cloud formations or floating objects). The cross-correlation metric will not show significance for images with uniform sky and without floating objects

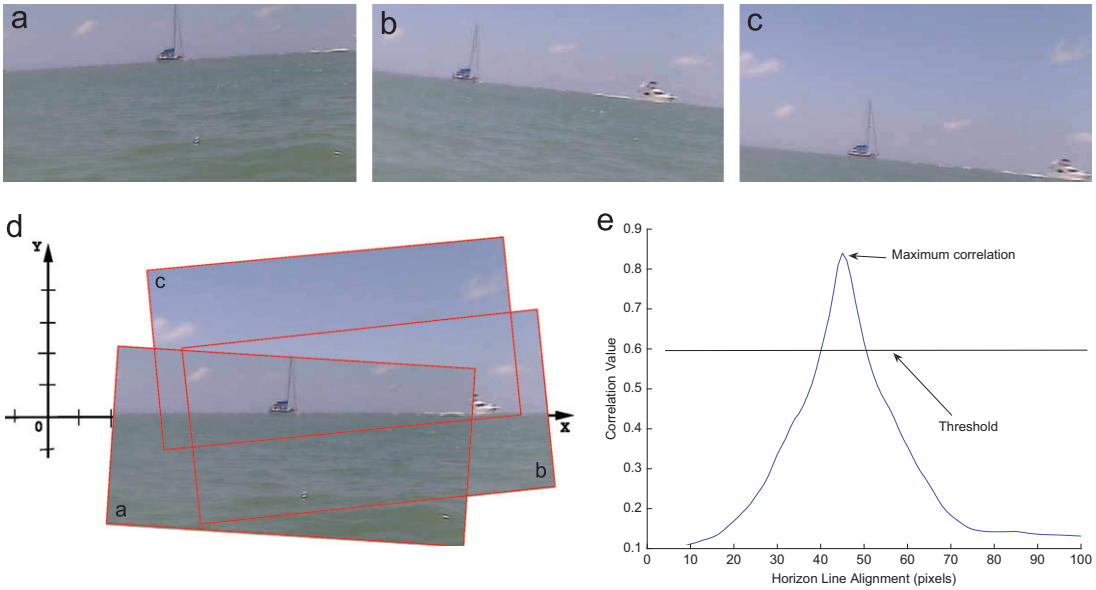

Fig. 4. Consecutive frames of a video sequence exhibit high magnitude non-linear intra-frame motion in (a)–(c) which is the result of rapid camera motion. (d) shows registration of (a)–(c) in one coordinate system. (e) shows the values of one-dimensional normalized cross-correlation scaled in the range [0..1] during registration of frames (a) and (b) along the horizon line. The peak corresponds to the optimal alignment along the horizon line.

(including ships) on the horizon, thus, halting further processing of those images.

3.4. Segmentation

The segmentation block of the algorithm identifies the regions of the image that potentially contain ships. The input to the block is a real-valued gradient image from the registration step and parameters of the horizon line. A strip of the input image of M pixels high that is adjacent to the horizon is used for the segmentation step. The segmentation is performed through global thresholding on such strip and results in a binary map. The value for a threshold is chosen by the method of Otsu (1979), which maximizes the between-class variance of foreground and background in the histogram. The value of a threshold is compared to minimum value threshold $T3$, which is learned on a separate dataset of images of clear sky and without ships. This eliminates the false positives originating from compression artifacts and natural gradient attributed to the sky appearance.

After the binary image of the foreground and background is obtained, the result is refined by eliminating spurious foreground objects caused by other factors: sea waves and curvature of the horizon line due to the radial distortion of the camera. Foreground pixels adjacent to the horizon line which result in projection profile (O'Gorman and Kasturi, 1995) of 20% or more of length of the horizon are disregarded from consideration.

The next filtering step evaluates the dimensions of the remaining connected objects. The component labeling algorithm (O'Gorman and Kasturi, 1995) is used to find connected foreground objects and Principal Component Analysis is applied to each of those objects to find their principal axes. The objects that are less than N pixels wide along the second largest eigen-vector are filtered out. The parameter N defines the sensitivity of our approach to the smallest objects.

All the remaining foreground objects are checked for proximity to the horizon line. The foreground objects that are close to the horizon are outputted as a result of the segmentation step. In the case that the foreground objects are not found, the image is disregarded from further processing. Fig. 5 shows the sequence of steps in the segmentation algorithm. The values of parameters K, L, M, N for our application are described in Section 5.

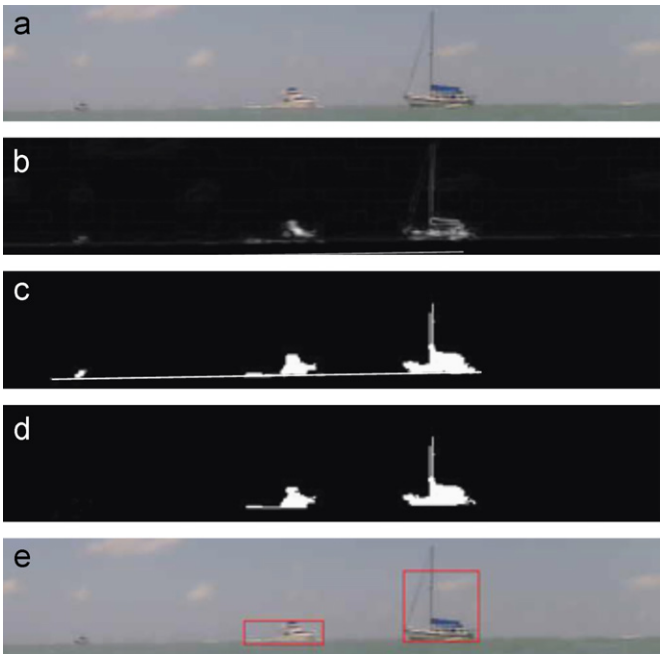

Fig. 5. Steps of the segmentation algorithm: (a) original image; (b) color-gradient filter; (c) thresholding; (d) filtering; (e) output.

3.5. Tracking

The tracking algorithm relies on a Multiple-Hypothesis Tracking (MHT) framework (Reid, 1979) which allows generation of multiple hypotheses for a track that can be later resolved with future observations. Each hypothesis is managed by the linear Kalman filter (Kalman, 1960) that provides a prediction step for possible location of targets in following frames of the video sequence. In our implementation of MHT algorithm up to two hypotheses are generated for each observation: existing tracks are projected forward to the time of observations and the nearest two (in terms of euclidian distance) tracks are assigned to two hypotheses. The depth of such parallel hypotheses tree-branch is three frames.

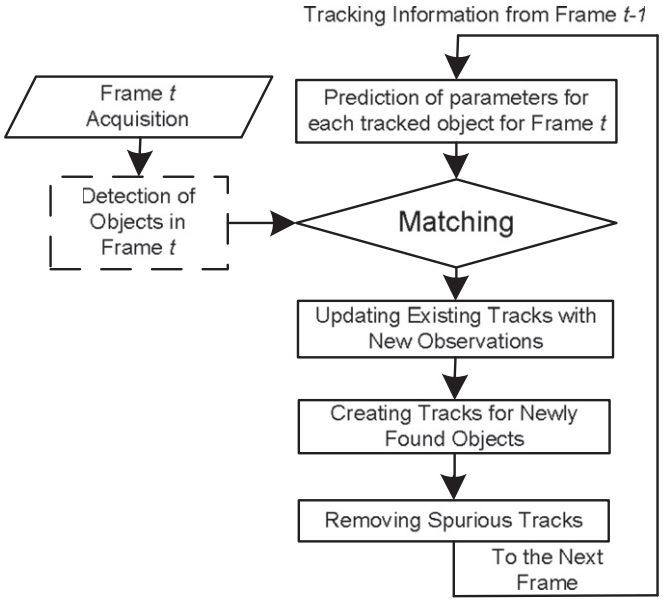

Fig. 6. Algorithm of tracking of detected objects throughout video sequence.

Such depth is enough to resolve ambiguity of data association between tracks and new observations for our dataset.

Two different filters are defined to track the following two entities: the position of the center of the bounding box (centroid) and the bounding box dimensions. The auxiliary variables for each filter describe the speed of change of the main variables. The tracking algorithm operates in the coordinate system global for all frames. This coordinate system is anchored to the horizon line.

A track for a new object is initiated if: (1) the object was found outside of validation regions of existing tracks, (2) if it was detected in two consecutive frames and the bounding boxes of such object in these two frames intersect. The values for the state and measurement vectors of the Kalman filter for that track are initiated from these two frames. The track for an object is terminated if the number of valid detections in the history of the track is less than half of the number of consecutive frames for which the track exists. The objects with the track length of more than 10 frames are considered marine vehicles and are provided as the system output. The outline of the tracking algorithm is summarized in Fig. 6.

4. Performance evaluation

The purpose of this section is to describe the methods of performance evaluation for the developed algorithm and its comprising components. First, several performance metrics are used to evaluate the horizon detection performance. Next, for detection of ships on a frame and sequence levels we use several metrics that account for many important measures of system performance such as number of objects detected and missed, false positives, fragmentation in a temporal dimension, and localization error of detected objects. Finally, we evaluate the impact of the tracking and registration steps of the algorithm on the performance of ship detection. To qualitatively evaluate the horizon and ship detections, three *evaluation* thresholds $ET1-ET3$ are introduced. Those thresholds are intended for performance evaluation only and are not part of the algorithm like thresholds $T1-T3$. The evaluation is conducted experimentally on a number of datasets which are described in Section 5.

4.1. Performance of horizon presence detection algorithm

This set of metrics is intended to measure the precision and recall of horizon status detection in images. When the horizon is present in an image, the image is labeled as 'horizon-image'. When the horizon is not present or when it is present but is obscured, blurred, or uncertain, the image is labeled as 'no-horizon-image'. The following are the metrics used to evaluate detection of horizon presence:

$$A_1 = \frac{tp}{tp+fp}, \quad A_2 = \frac{tp}{tp+fn} \tag{10}$$

where A_1 is the precision of detection, A_2 is the recall, tp,fp,fn are true positive, false positive, and false negative outcomes of the classification correspondingly.

4.2. Performance of horizon localization algorithm

Localization of the horizon is the process of determining the parameters of the horizon line in those images where the horizon is present. In order to evaluate such performance the algorithm's output is compared with the ground-truth using two performance metrics to reflect different aspects of performance. Both metrics use pixelwise comparison of the candidate and target data. The first metric represents the percentage of pixels in all images correctly separated into the water and sky regions by the found horizon line:

$$A_3 = \frac{1}{k}\sum_{i=1}^{k}\frac{N_c^i}{N_i} \tag{11}$$

where k is the number of images in the dataset, N_c^i-number of pixels correctly separated by the found horizon line in image i, N_i - number of pixels in the image i. This metric provides a reference to the general performance of the horizon localization algorithm on a dataset of images.

The second metric reflects performance of horizon localization on each of the images of the dataset. The horizon line in an image is considered detected correctly if the percentage of pixels in the image correctly separated by the horizon line is above the specified evaluation threshold. The percentage of such images in the dataset will define the second accuracy metric:

$$A_4 = \frac{n_c}{k} \tag{12}$$

where k is the total number of images in the dataset and n_c is the number of images in the dataset where the horizon line was detected with accuracy above the given threshold:

$$\frac{N_c^i}{N_i} \geq ET1 \tag{13}$$

where $ET1$ is the evaluation threshold value.

4.3. Performance of ship detection and localization

The performance of ship detection is analyzed in two stages. In the first stage, a frame-wise localization and matching is done between the ground-truth targets and detected objects. A decision for each ground-truth target in every frame is made: detected or missed. The false positives are also counted. In the second stage of evaluation a decision is made on a sequence level: the target is detected in the sequence, missed, or a false alarm occurred.

The spatial localization of ground-truth targets and detected objects is performed using rectangular bounding boxes. For simplicity of representation, a bounding box around targets and detected objects is always oriented so that its sides are parallel to

the axes of the image plane. To facilitate creation of the ground-truth, the ViPER tool by Doermann and Mihalcik (2000) is used to create bounding boxes around targets in the data. The spatial overlap between matched ground-truth targets and detected objects is computed according to several metrics.

The precision and recall of localization for an object in a frame is computed according to the following relations:

$$PrecisionLoc = \frac{G \cap D}{G}, \quad RecallLoc = \frac{G \cap D}{D} \tag{14}$$

where G is a area of the bounding box under the ground-truth target, D – area of the bounding box under the detected object matched to its ground-truth, $G \cap D$ – spatial overlap between the matched ground-truth target and detected object.

The dice coefficient (Doermann and Mihalcik, 2000) is computed as follows:

$$Dice = \frac{2(G \cap D)}{G+D} \tag{15}$$

The dice metric avoids the asymmetry of recall and precision, and thus, is better suited for detection on a higher levels of analysis. The recall and precision of localization are only reported on the frame-level. All three metrics are normalized between 0 and 1.

In order to decide if the object is detected in the frame, missed, or a false alarm occurred, the localization according to the dice overlap (15) for each ground-truth target in the frame is compared to a pre-defined frame-level evaluation threshold $ET2$.

The second stage of evaluation analyzes detection of ships on the sequence level. From the first stage of evaluation, a particular target may be detected in its sequence for a certain number of frames. If the set of frames where the target is detected overlaps the appropriate number of frames where the ground-truth for it exists, the target is declared detected in the sequence. Otherwise, the target is declared missed. The overlap on the sequence level is computed according to the temporal dice coefficient, which is compared to the sequence-level evaluation threshold $ET3$:

$$TemporalDice = \frac{2(F_G \cap F_D)}{F_G+F_D} \tag{16}$$

where F_G is the set of frames where the ground-truth for the target exists, F_D – set of frames where the object is detected, $F_G \cap F_D$ – temporal overlap between the detected object and its ground-truth in the sequence, i.e. the set of frames where the detected object and its ground-truth exists and they are spatially matched in the first stage of analysis.

By using the temporal dice coefficient (16) the counts of targets detected, targets missed, and false positives are found for each sequence. The precision of detection and recall of detection are, then, calculated for all targets in all sequences. The precision and recall of detection in all the sequences is defined as follows:

$$PrecisionSeq = \frac{C_D}{C_G}, \quad RecallSeq = \frac{C_D}{C_D+C_F} \tag{17}$$

where C_D is the count of objects in all sequences that are correctly detected, C_G – count of all ground-truth objects, C_F – count of all false positives

These two metrics are used to report results of detection of all targets across all the sequences provided evaluation thresholds $ET2$ and $ET3$.

4.4. Evaluation of impact of registration and tracking steps

The registration and tracking steps of the algorithm are indirectly evaluated for their impact on ship detection performance when modifying the pipeline in Fig. 2 to include only some steps of the algorithm according to several experimental setups. These setups are as follows:

1. The ship-detection pipeline includes only horizon detection and segmentation steps.
2. The ship detection pipeline includes horizon detection, segmentation, and tracking steps.
3. The ship detection pipeline includes all steps: horizon detection, registration, segmentation, and tracking.

Such evaluation is performed according to metrics (14)–(17) under fixed settings of evaluation thresholds $ET2$ and $ET3$.

In this work, the direct evaluation of the tracking accuracy is not performed due to the absence of identification/recognition of targets which are required to evaluate accuracy of generation of correct trajectories of ships. That is, we are not evaluating ID switches in correspondences between the ground-truth and detections. Neither we evaluate the degree of fragmentation of detection tracks directly. We also do not directly evaluate the accuracy of the image registration performance due to lack of the ground-truth. Both the registration and tracking steps are primarily used to improve precision of ship detection and reduce false alarms.

5. Results

The prototype of the system described by Fefilatyev et al. (2009) was used to collect a large dataset of video sequences. The system was tested in several locations off the Florida's coastline during six deployments. The video was taken in various weather conditions which included clear sky, partially clouded sky, and overcast during day-time. The deployment locations included the open ocean, as well as in littoral areas when the camera was pointing away from the shore. The video, which was recorded in a resolution of 800×600 at the rate of 10 frames per second, had significant compression artifacts due to storage in MJPEG format. A subset of the video data, called *Performance Evaluation dataset*, was annotated for performance evaluation. The dataset consisted of 550 video sequences selected randomly from all available data (will be available on our web-site). Each sequence consisted of 100 frames representing a contiguous 10-s interval. Instances of ships within a video sequence included single and multiple vessels (see Fig. 7), some entering and leaving the field of view, as well as periods with no ships in the view. All 550 sequences were annotated with ground-truth ships by using ViPER tool. The other datasets used in the training and evaluation of the algorithm are as follows:

- *Development dataset* of 150 images divided into three subsets: (a) A subset to learn the value of the threshold parameter $T1$

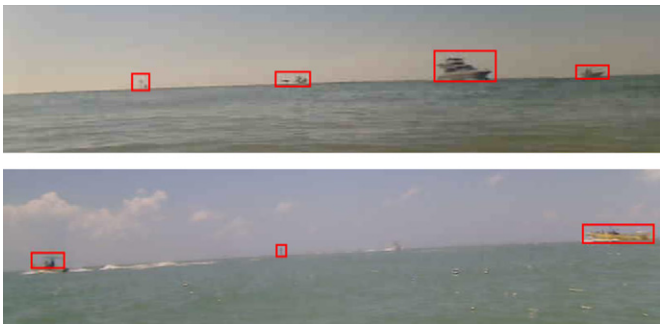

Fig. 7. Tracking multiple targets from a camera installed on a buoy. The buoy is deployed near a channel.

for horizon anomaly detection. The subset consisted of 50 images outside of the data collected from the prototype. The images were split into two categories: 'horizon-present' and 'horizon-not-present'. (b) A subset to learn the value of the threshold parameter $T2$ for registration validation. The subset consisted of 50 image of the dataset from the prototype split into two categories: 'clear-sky' and 'textured-sky' to mark the images where the area above the horizon was uniform (no clouds, no floating objects) and images with non-uniform sky. (c) A subset to learn the value of the threshold parameter $T3$ for segmentation validation. The subset consisted of 50 images split into two categories: 'clear-horizon' and 'ships-horizon' to mark the images where the horizon did not have any objects of interest and the images having ships on the horizon.

- *Horizon presence evaluation dataset*: The dataset was created to evaluate performance of horizon presence detection and consisted of approximately 10% the images of the annotated dataset. Those images where the horizon was not present were marked as 'no-horizon'. All other images in the dataset were considered 'horizon'-images. The majority of the images in the dataset contained horizon.
- *Horizon localization evaluation dataset*. The dataset was created to evaluate the accuracy of horizon line localization and consisted of 150 images with annotated water and sky regions.

Table 1 summarizes the datasets and Table 2 shows the values of the non-threshold parameters used in different steps of the algorithm.

5.1. Performance of horizon presence detection algorithm

Approximately 95% of the images in the *Horizon presence evaluation dataset* contained horizon and belonged to the category 'horizon' images. 'No-horizon' category were represented by 5% of images, where 4% of the images contained water regions and 1% of the images contained only sky regions. Table 3 shows the results of classification of the dataset according to metrics $A1$, $A2$ (10) into those two categories.

The accuracy and recall for the 'horizon' category is achieved at 98.1% and 99.6% correspondingly. For the 'non-horizon' category those figures are at 88.0% and 57.9%. The algorithm is biased toward the 'non-horizon' category, but since we are more interested in those images which potential have ships this performance is acceptable.

Table 1
Datasets used to train and evaluate the algorithms.

Description	# Images
Performance evaluation dataset	55 000
Development dataset	150
Horizon presence evaluation dataset	5400
Horizon localization evaluation dataset	150

Table 2
Values of parameters used during performance evaluation.

Parameter	Description	Value
K	Number of candidate lines for horizon consideration	5
L	Height of rectangular patches used for image registration, pixels	75
M	Height of rectangular patches over horizon used for segmentation, pixels	40
N	Minimum width of the object along the second largest eigen-vector, pixels	6
α	Rate of update for registration map	0.05

Table 3
Results of performance evaluation of horizon presence detection algorithm.

	Actual 'Horizon' images	Actual 'No-horizon' images
Classified as 'Horizon'	5148	98
Classified as 'No-horizon'	19	135

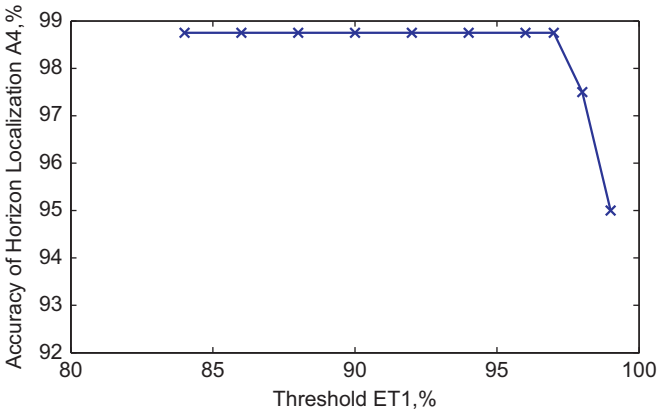

Fig. 8. Accuracy of horizon localization.

5.2. Performance of horizon localization

The accuracy of horizon localization on the *Horizon Localization Evaluation dataset* according to the $A3$ metric (11) is 99.26%. Fig. 8 shows the value of accuracy horizon localization according to metric $A4$ (12) for a number of evaluation threshold values $ET1$. Overall, the accuracy of localization is high for both metrics. The metric (12) sees degradation of performance with the threshold of 98% and above. However, such threshold is very close to 100% and the human error during the process of creating ground-truth may be the primary factor resulting in such performance.

5.3. Performance of ship detection

5.3.1. Localization on frame level

The performance of localization on a frame level is computed according to metrics (14) by averaging the results from each frame in all sequences and is a general metric for qualitative spatial overlap between the ground truth and found targets, not the fraction of ships detected. The precision of localization is 76% with lower score for recall – 61%. According to our observations, such recall performance is observed because of the high number of false positives that occur due to artifacts in MJPEG data, especially small objects with height around 6 pixels that occur during rapid erratic motion of the camera attached to the buoy.

5.3.2. Detection on sequence level

The ship detection precision and recall of the algorithm on a sequence level can be shown according to three evaluation parameters. The first parameter, the evaluation threshold $ET2$, corresponds to the detection of a ship on a frame level depending on the value of overlap between the ground truth and found objects. Once a hard decision (i.e. detected or not) is made in each

frame regarding each target, it is possible to evaluate detection on a sequence level. The detection according to second evaluation parameter, the threshold *ET*3, produces a hard decision for detection of a target in a sequence. The third evaluation parameter, minimum height of objects, is needed to evaluate performance according to the size of the acceptable object. Thus, the measure of performance may vary greatly depending on the combination of evaluation parameters, although the performance of the algorithm itself does not depend on the evaluation parameters. In this work we show performance only for a subset of those evaluation combinations.

Table 4 and Fig. 9 show performance of detection for all targets on all the sequences according to precision and recall metrics (17) with different values of temporal threshold *ET*3 with the localization threshold *ET*2 fixed on the level of 0.5 and with the minimum height set to 6 pixels. The value 0.5 is the most natural choice for a symmetric metric such as dice. The algorithm performed well even on the values of the temporal threshold *ET*3 of up to 0.7. Up to 88% ships are detected in all sequences when at least small temporal overlap between ground truth and system's output is counted. According to our observations, misses and false positives usually occur due to small objects, with object-heights close to the minimum height of 6 pixels. In our dataset we have a substantial number of ships with sizes in vertical dimension close to that minimum. The detection recall for objects with heights larger than 6 is higher in those frames where the horizon is detected correctly. For example, Table 5 and Fig. 10 show the dynamics of performance when evaluation threshold values for *ET*2 and *ET*3 are fixed at 0.5 and when the ground-truth and system output are filtered by minimum height of the objects.

Table 4
Value of precision and recall of detection on the sequence level according to various values of evaluation threshold ET3, with threshold ET2 fixed at 0.5.

Threshold ET3	Precision (%)	Recall (%)
0.1	88.63	72.76
0.2	83.89	68.87
0.3	82.46	67.70
0.4	80.57	66.15
0.5	75.83	62.26
0.6	72.04	59.14
0.7	66.82	54.86
0.8	49.29	40.47
0.9	22.27	18.29
1.0	2.84	2.33

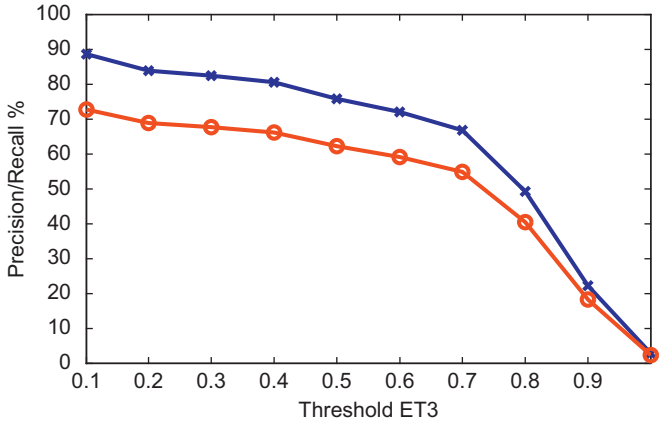

Fig. 9. Precision and recall of detection on the sequence level according to various values of threshold ET2, with threshold ET3 fixed at 0.5.

Table 5
Value of precision and recall of detection on the sequence level according to various values of minimal height of objects, with thresholds ET2 and ET3 fixed at 0.5.

Minimum height	Precision (%)	Recall (%)
6	75.83	62.26
7	75.83	62.50
8	75.83	65.57
9	75.36	68.24
10	75.36	70.98
11	75.36	72.27
12	70.73	71.82

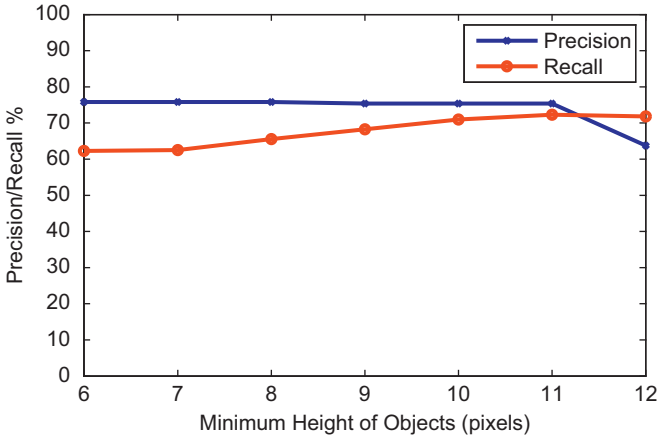

Fig. 10. Precision and recall of detection on the sequence level according to various minimum heights of objects, with evaluation thresholds ET2 and ET3 fixed at 0.5.

Table 6
Impact of image registration and tracking steps on ship detection performance.

Performance metric	H+S (%)	H+S+T (%)	H+R+S+T (%)
PrecisionLoc (14)	84.3	77.2	76.0
RecallLoc (14)	29.9	36.5	71.0
PrecisionSeq (17)	80.1	76.4	75.6
RecallSeq (17)	33.1	37.0	62.2

5.4. Impact of registration and tracking steps on ship detection performance

The impact of the tracking and registration steps on the performance of the algorithm is shown in Table 6. Three setups are evaluated according to several metrics described in Section 4.3. In the first setup, the algorithm's pipeline described in Section 3.1 (see Fig. 2) includes only the horizon detection and segmentation steps (encoded as "H+S" column in the table). Thus, the segmentation step provides the final output of the algorithm. In the second setup, the algorithm's pipeline includes the horizon detection, segmentation, and tracking steps (encoded as "H+S+T" column). In the third setup, the algorithm's pipeline includes all the steps: the horizon detection, registration, segmentation, and tracking (encoded "H+R+S+T"). Evaluation of precision and recall on the sequence level is performed with the *ET*2 and *ET*3 thresholds fixed on the level 0.5.

As it seen from the table the registration and tracking steps significantly improve the recall-type performance on both frame and sequence level and slightly reduce the precision-type performance. Such improvement for recall is explained by the filtering

step within the tracking algorithm which removes the outlier-detections that may show up after the segmentation step due to compression artifacts in the video. The effect of tracking on the precision accuracy has its benefits and drawbacks. Since the tracking algorithm may tolerate small misses in detection by using the prediction step of the Kalman filter it improves the performance for a sequence of frames which has small 'gaps' in detection. However, in many cases some valid detections from the segmentation step may be filtered out by the tracking algorithm if they show intermittent history. As it is seen from the table, the registration step is essential for improvement of recall performance when used along with tracking. This can be explained by the fact that the tracking algorithm by itself is not very useful in the video with high magnitude non-linear inter-frame motion, but becomes simple and efficient when tracking is performed along the horizon line in the global coordinate system.

5.5. Discussion

Overall, scheme of detection, proved to be robust with reliable horizon detection, frame registration, segmentation and tracking of targets. The visual results obtained on the data from the prototype outside the annotated 550 sequences showed good performance as well. It is important to note that the data in the collected dataset was 'difficult' not only because of the non-stationary camera, but because of the abundance of compression artifacts due to the low-quality imaging sensor present in the prototype of the system. It is expected that for the imagery not affected by the mentioned artifacts the performance of the system will be better, especially in relation to the false positives.

The current algorithm does not assume correct registration of images in the physical world. There are at least two cases that can negatively affect the registration of a frame in the global coordinate system. First, if the above-the-horizon area of the image does not contain any texture like clouds or floating objects the registration is not possible as there is no reference. Another anomaly case occurs in frame with a single above-the-water feature represented by a moving object (either fast moving ship or clouds). In such a case the global coordinate system may become locked on such object and not on the surrounding physical environment. However, the main application of the registration step is to aid the consequent tracking which greatly impacts the ship detection results as demonstrated in Table 6.

In this work, we did not evaluate directly the accuracy of tracking step due to the absence of the identification/recognition of targets. Introduction of such step based on appearance would allow tracking of individual ship trajectories. This is one of the directions of our further research.

The current assumption of open sea environment is critical for the algorithm's performance. A useful expansion of the current work would be a possibility of detection of ships in front of coastal lines. Such a problem would require a different approach to horizon detection and segmentation.

6. Conclusions

This paper presents novel algorithms for open-sea visual maritime surveillance using a highly non-stationary camera. The camera installed on a buoy is a subject to rapid erratic motion. The proposed algorithm detects, localizes, and tracks ships in the field of view of the camera and outputs images of the found targets. The experiments, conducted on a large dataset of video data obtained from a prototype, show good results. Specifically, the algorithm detects and tracks correctly of up to 88% of ships. In the context of ship detection, a new horizon detection scheme was developed for a complex maritime domain that provides accuracy of horizon localization of 98%, and detection of horizon images with the accuracy rate of 99%. The developed algorithms are fast and are well suited for low-powered autonomous systems deployed for long periods of time. This work addresses the visual surveillance in open sea: the horizon line around the platform does not contain any coastline or objects of infrastructure. A future research will focus on a more complex situation: detection and tracking of ships in coastal regions, when coastal line is visible and may include profiles of the ships.

Acknowledgment

This work was partially supported by the USF Computational Tools for Discovery Thrust, USF Graduate Multidisciplinary Scholars Program, and by funding provided by the Office of Naval Research.

References

Araghi, L., Khaloozade, H., Arvan, M., 2009. Ship identification using probabilistic neural networks (PNN). In: Proceedings of International MultiConference of Engineers and Computer Scientists, vol. 2, pp. 18–20.

Bao, G., Xiong, S., Zhou, Z., 2005. Vision-based horizon extraction for micro air vehicle flight control. IEEE Trans. Instrum. Meas. 54 (3), 1067–1072.

Benetazzo, A., 2011. Accurate measurement of six degree of freedom small-scale ship motion through analysis of one camera images. Ocean Eng. 38 (16), 1755–1762.

Bhattacharyya, A., 1943. On a measure of divergence between two statistical populations defined by their probability distributions. Bull. Calcutta Math. Soc. 35, 99–109.

Bloisi, D., Iocchi, L., Remagnino, P., Monekosso, N., 2009. Argos–a video surveillance system for boat traffic monitoring in Venice. Pattern Recognition Artif. Intell. 23 (7), 1477–1502.

Cao, H., Zhang, J., 2007. Video stabilizing and tracking by horizontal line for maritime cruise ship. In: Proceedings of International Conference on Control and Automation, pp. 1202–1206.

Cao, X., Wu, L., Rasheed, Z., Liu, H., Choe, T., Guo, F., Haering, N., 2010. Automatic geo-registration for port surveillance. Int. J. Pattern Recognition Artif. Intell. 24 (4), 531–555.

Chen, H., 2008. Vision-based tracking with projective mapping for parameter identification of remotely operated vehicles. Ocean Eng. 35 (10), 983–994.

Criminisi, A., Zisserman, A., 2000. Shape from texture: homogeneity revisited. In: Proceedings of British Machine Vision Conference, pp. 82–91.

Di Zenzo, S., 1986. A note on the gradient of a multi-image. Comput. Vision Graph. Image Process. 33 (1), 116–125.

Doermann, D., Mihalcik, D., 2000. Tools and techniques for video performance evaluation. In: Proceedings of International Conference on Pattern Recognition, vol. 4, pp. 167–170.

Dusha, D., Boles, W., Walker, R., 2007. Fixed-wing attitude estimation using computer vision based horizon detection. In: Proceedings of Australian International Aerospace Congress, pp. 1–19.

Enríquez de Luna, Á., Miravet, C., Otaduy, D., Dorronsoro, C., 2005. A decision support system for ship identification based on the curvature scale space representation. In: Proceedings of SPIE, vol. 5988, pp. 171–182.

Ettinger, S., Nechyba, M., Ifju, P., Waszak, M., 2003. Vision-guided flight stability and control for micro air vehicles. Adv. Robotics 17 (7), 617–640.

Fefilatyev, S., Goldgof, D., 2008. Detection and tracking of marine vehicles in video. In: Proceedings of International Conference on Pattern Recognition, pp. 1–4.

Fefilatyev, S., Goldgof, D., Lembke, C., 2009. Autonomous buoy platform for low-cost visual maritime surveillance: design and initial deployment. In: Proceedings of SPIE, vol. 7317, p. 73170A.

Fefilatyev, S., Goldgof, D., Lembke, C., 2010. Tracking ships from fast moving camera through image registration. In: Proceedings of International Conference on Pattern Recognition, pp. 3500–3503.

Feineigle, P., Morris, D., Snyder, F., 2007. Ship recognition using optical imagery for harbor surveillance. In: Proceedings of Association for Unmanned Vehicle Systems International (AUVSI).

Guo, F., Chellappa, R., 2006. Video mensuration using a stationary camera. In: Lecture Notes in Computer Science, vol. 3953, pp. 164–176.

Gupta, K., Aha, D., Moore, P., 2009. Case-based collective inference for maritime object classification. In: Proceedings of International Conference on Case-Based Reasoning, pp. 434–449.

Kalman, R., 1960. A new approach to linear filtering and prediction problems. J. Basic Eng. 82 (1), 35–45.

Kruger, W., Orlov, Z., 2010. Robust layer-based boat detection and multi-target-tracking in maritime environments. In: Proceedings of International Waterside Security Conference, pp. 1–7.

Lemoine, G., Schwartz-Juste, G., Kourti, N., Shepherd, I., Cesena, C., 2005. An open source framework for integration of vessel positions detected in spaceborne SAR imagery in operational fisheries monitoring and control. In: Proceedings of ENVISAT/ERS Symposium, vol. 572, pp. 252–258.

Li, H., Wang, X., 2008. Automatic recognition of ship types from infrared images using support vector machines. In: Proceedings of International Conference on Computer Science and Software Engineering, vol. 6, pp. 483–486.

Li, J., Liu, J., Tian, J., 2004. Infrared image segmentation via intelligent genetic algorithm based on 2-D maximum fuzzy entropy. In: Proceedings of International Conference on Advances in Dynamics, Instrumentation and Control, pp. 414.

Liu, H., Javed, O., Taylor, G., Cao, X., Haering, N., 2008. Omni-directional surveillance for unmanned water vehicles. In: Proceedings of International Workshop on Visual Surveillance.

Lv, F., Zhao, T., Nevatia, R., 2006. Camera calibration from video of a walking human. IEEE Trans. Pattern Anal. Mach. Intell. 28 (9), 1513–1518.

McGee, T., Sengupta, R., Hedrick, K., 2005. Obstacle detection for small autonomous aircraft using sky segmentation. In: Proceedings of International Conference on Robotics and Automation, pp. 4679–4684.

Morris, D., Colonna, B., Snyder, F., 2007. Image-based motion stabilization for maritime surveillance. In: Proceedings of SPIE, vol. 6497, pp. 64970F–1.

O'Gorman, L., Kasturi, R., 1995. Document Image Analysis. IEEE Computer Society Press Los Alamitos, CA, USA.

Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. 9 (1), 62–66.

Reid, D., 1979. An algorithm for tracking multiple targets. IEEE Trans. Autom. Control 24 (6), 843–854.

Rodriguez Sullivan, M., Shah, M., 2008. Visual surveillance in maritime port facilities. In: Proceedings of SPIE, vol. 6978, pp. 29.

Sanderson, C., Gibbins, D., Searle, S., 2008. On statistical approaches to target silhouette classification in difficult conditions. Digital Signal Process. 18 (3), 375–390.

Saur, G., Estable, S., Zielinski, K., Knabe, S., Teutsch, M., Gabel, M., 2011. Detection and classification of man-made offshore objects in terrasar-x and rapideye imagery: selected results of the demarine-deko project. In: Proceedings of IEEE-Spain OCEANS, pp. 1–10.

Seibert, M., Rhodes, B., Bomberger, N., Beane, P., Sroka, J., Kogel, W., Kreamer, W., Stauffer, C., Kirschner, L., Chalom, E., et al., 2006. SeeCoast port surveillance. In: Proceedings of SPIE, pp. 62040B–1.

Shaik, J., Iftekharuddin, K., 2009. Detection and tracking of targets in infrared images using Bayesian techniques. Opt. Laser Technol. 41 (6), 832–842.

Sheikh, Y., Haering, N., Shah, M., 2006. Shape from dynamic texture for planes. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2285–2292.

Socek, D., Culibrk, D., Marques, O., Kalva, H., Furht, B., 2005. A hybrid color-based foreground object detection method for automated marine surveillance. In: Advanced Concepts for Intelligent Vision Systems, vol. 3708, pp. 340–347.

Tao, W., Jin, H., Liu, J., 2007. Unified mean shift segmentation and graph region merging algorithm for infrared ship target segmentation. Opt. Eng. 46 (12), 127002.

Teutsch, M., Kruger, W., 2010. Classification of small boats in infrared images for maritime surveillance. In: Proceedings of International Waterside Security Conference, IEEE, pp. 1–7.

Valin, P., Rhéaume, F., Tremblay, C., Grenier, D., Jousselme, A., Bossé, E., 2006. Comparative implementation of two fusion schemes for multiple complementary FLIR imagery classifiers. Inf. Fusion 7 (2), 197–206.

Voles, P., Teal, M., Sanderson, J., 1999. Target identification in a complex maritime scene. In: IEE Colloquium on Motion Analysis and Tracking (Ref. No. 1999/103), IET, pp. 1–15.

Wei, H., Nguyen, H., Ramu, P., Raju, C., Liu, X., Yadegar, J., 2009. Automated intelligent video surveillance system for ships. In: Proceedings of SPIE, vol. 7306, pp. 73061N.

Zhang, H., Yin, P., Zhang, X., Shen, X., 2010. A robust adaptive horizon recognizing algorithm based on projection. Trans. Inst. Meas. Control 33 (6), 734–751.

Zitova, B., Flusser, J., 2003. Image registration methods: a survey. Image Vision Comput. 21 (11), 977–1000.