Enhancing Automatic Maritime Surveillance Systems with Visual Information

Domenico D. Bloisi, Fabio Previtali, Andrea Pennisi, Daniele Nardi, Michele Fiorini, *Senior Member, IEEE*

***Abstract*—Automatic surveillance systems for the maritime domain are becoming more and more important due to a constant increase of naval traffic and to the simultaneous reduction of crews on decks. However, available technology still provides only a limited support to this kind of applications. In this paper, a modular system for intelligent maritime surveillance, capable of fusing information from heterogeneous sources, is described. The system is designed to enhance the functions of the existing Vessel Traffic Services systems and to be deployable in populated areas, where radar-based systems cannot be used due to the high electromagnetic radiation emissions. A quantitative evaluation of the proposed approach has been carried out on a large and publicly available data set of images and videos, collected from multiple real sites, with different light, weather, and traffic conditions.**

***Index Terms*—video analysis, infrared surveillance, object detection, sensor fusion.**

I. Introduction

Automatic surveillance of coastal areas is gaining importance due to the increasing global ship traffic: Tankers, container ships, and bulk carriers are the most important means of transportation of our time [1]. The simultaneous reduction of crews on decks makes the adoption of automatic tools a necessary requirement for port management. Moreover, the presence of environment protection issues and new dangerous threats coming from the sea, including illegal smuggling and fishing, immigration, oil spills and piracy, encourage the development of intelligent monitoring systems.

A possible strategy to develop a robust maritime surveillance solution is to gather and merge data from multiple heterogeneous sensors [2]. Examples are systems combining Automatic Identification System (AIS) data with Synthetic Aperture Radar (SAR) imagery (e.g., [3]), buoy-mounted sensors with land radars (e.g., [4], [5]), visual- with radar-based surveillance (e.g., [6]) and multiple ship-based sensors - e.g., [7].

In this paper, we focus on Vessel Traffic Services (VTS) systems, which combine radar and AIS data and are often equipped with long-range surveillance cameras, both electro-optical (EO) and infra-red (IR). However, using radar and AIS data only is not sufficient to ensure a complete solution for the maritime surveillance problem, due to two strong limitations:

1) AIS signal may be not available (AIS device not activated or malfunctioning) or illegally manipulated
2) radar-based systems are not suitable for vessel traffic monitoring in populated areas, due to the high electromagnetic radiation emissions.

Replacing radar sensors with cameras is a feasible solution for the maritime surveillance task, without the need of placing radar antennas in populated areas [8]. Here, we propose a modular architecture that extends the capability of currently available VTS systems, together with a prototype system that allows adds a novel visual dimension to the common VTS features. The architecture is designed for:

1) detecting boats through a classifier-based method, which can work with both EO and IR moving cameras.
2) tracking multiple ships, even in presence of occlusions
3) fusing data from existing VTS systems with visual information from cameras
4) deployable in populated areas

The main contribution of this paper is to provide a general scheme with a suitable approach for combining AIS and visual (EO and IR) data in a unique view. The system resulting from the implementation of the scheme has been quantitatively evaluated on real and publicly available[1] data coming from different currently working VTS systems.

The rest of this paper is organized as follows. Related work is discussed in Section II, while the system overview is given in Section III. Vessel detection with moving and zooming cameras is detailed in Section IV; tracking and data fusion are described in Section V. The experimental results are shown in Section VI. Finally, conclusions are drawn in Section VII.

II. Related Work

Surveillance systems for the maritime domain have to deal with a set of challenges, including:

- wide areas to be monitored
- weather issues, such as heavy rain or fog
- night-time monitoring, with the need of suitable sensors
- varying size (ranging from few to hundreds of meters in length) of the relevant objects
- multiple objects to be tracked, with possible partial and total occlusions

In order to perform an accurate and effective monitoring of coastal areas, it is thus necessary to collect and combine data from multiple and heterogeneous information sources. Existing approaches addressing information fusion solutions for maritime surveillance can be grouped according to the

D.D. Bloisi, F. Previtali, and D. Nardi are with the Department of Computer, Control, and Management Engineering, Sapienza University of Rome, via Ariosto 25, 00185, Rome, Italy e-mail: ⟨lastname⟩@dis.uniroma1.it

A. Pennisi is with the Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, Pleinlaan, 2, B-1050 Brussels, Belgium e-mail: apennisi@etro.vub.ac.be

M. Fiorini is with Selex ES S.p.A - A Finmeccanica Company, Rome, Italy e-mail: michele.fiorini@selex-es.com

[1]All data used in this work can be found at http://goo.gl/jTYuTi

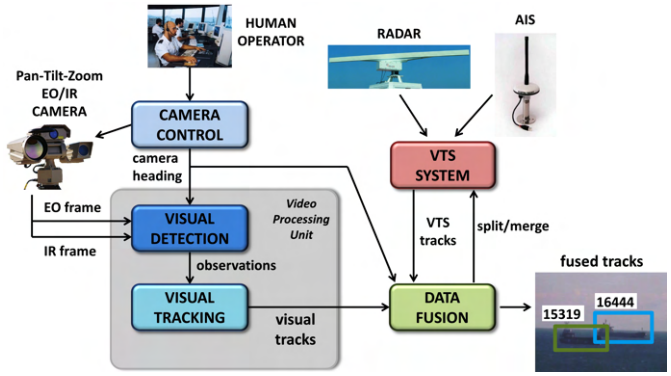


Fig. 1. Modular architecture of the system. The output is a unique view containing both VTS and visual data.

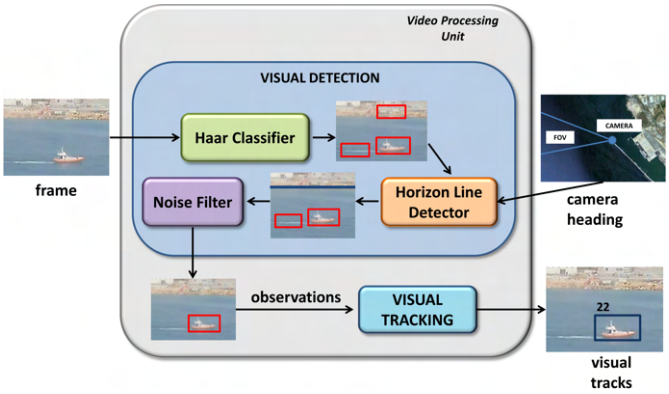


Fig. 2. Video Processing Unit. The detection is based on a Haar classifier, while the visual tracking module uses the PTracking library [17].

nature of the main sensors used: *(i)* radar, *(ii)* satellite, *(iii)* sonar, and *(iv)* camera. In this paper, we focus on systems where visual information is available, thus this section contains a discussion about camera-based solutions. Reviews about maritime surveillance systems relying on radar, satellite, and sonar sensors can be found in [9] and [10].

Camera-based systems. Bechar *et al.* [11] address the problem of near real-time video analysis of a maritime scene using a moving airborne RGB video camera for vessel detection. The approach uses a fusion of spatio-temporal uncertainty, which has been recast as an energy minimization problem. The main drawback is the use of multiple hard coded parameters, tuned according to the experience of the authors. Histogram of Oriented Gradients (HOG) are used in [12]. Since the calculation of the detection features involves a significant amount of computational resources, real-time performance can be obtained only adopting hardware acceleration with programmable components (e.g., FPGAs). Bousetouane and Morris [13] investigate the use of Convolutional Neural Networks (CNNs) for vessel classification. They found that OverFeat features outperform other state-of-art CNN architectures for a commercial fishing harbor data set. Qi *et al.* [14] propose an approach for fast detecting small maritime objects in IR images. The method is based on the local minimum patterns (LMP). Using a saliency map via background subtraction, potential objects are extracted by means of an automatically estimated threshold based on the LMP. It tooks 10 ms to process a 352×288 image.

Multiple sensor fusion. An open problem in the maritime scenario is how to fuse information from the different sensors. Bustamante *et al.* [15] propose a multi-agent system (MAS) architecture for automatically controlling the camera, radar, and AIS modules. The data fusion system is represented as an additional source sensor, which allows the agents to collaborate for avoiding redundancy. However, the conflict solving mechanism between agents can cause lost time. Marti *et al.* [16] propose a fusion system that combines information coming from the on-board sensor together with the messages from collaborative entities and static databases. Even if the approach can achieve good results, the participation of an external user is often needed for obtaining satisfactory performance during the fusion procedure.

In this paper, our aim is to describe an approach for enhancing the traditional VTS system functions with visual information. VTS systems provide information, denoted from now on as *VTS data*, combining radar and AIS tracks. To this end, we propose a camera-based architecture in which visual sensors (both EO and IR) can be used alone or in combination with VTS data to obtain a vessel traffic monitoring system with a high accuracy. The system is deployable both in a site where a VTS is already present and where it is not possible or not convenient to install a radar.

III. System Overview

The modules of the proposed architecture are shown in Fig. 1. An EO/IR camera is the main sensor and it can be moved by a human operator. The *Camera Control* module provides the current orientation and field-of-view (FOV) of the camera to the *Video Processing Unit* (VPU), which is responsible to detect and track the vessels using only visual information. Since the camera can move and zoom, the detection task is rather complex, since it is not possible to create a model of the observed scene.

The *Data Fusion* module receives data from both the VPU and the VTS system. Its role is to fuse the visual tracks coming from the video analysis with the tracks generated from an existing VTS system. In this way, it is possible to provide the user with a novel visual dimension in addition to the traditional geo-referenced, radar-like VTS view. Moreover, the Data Fusion module can send feedback information to the VTS system, in order to adapt the radar parameters for improving the detection accuracy (see [18] for details).

In the following section, the visual detection process is described, while the tracking and data fusion steps are discussed in Section V.

IV. Visual Detection

The Visual Detection module is part of the VPU and its aim is to find the objects of interest in the current input image. Since the detection accuracy affects all the stages in the VPU process flow, it must be as high as possible, while maintaining an acceptable computational load. The three main components

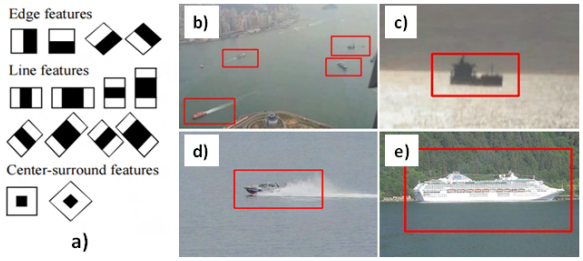


Fig. 3. a) Types of Haar-like features used for detection. b) Multiple boats. c) Reflections on the water surface. d) Boat wakes. e) Cruise ship. Pictures b) and c) are from MarDCT data set [24], while d) and e) are from the PASCAL VOC data set [20].

of the VPU, namely the Haar classifier, the horizon line detector, and the noise filter, are shown in Fig. 2.

Haar classifier. We assume that the camera can be freely moved by the user, thus a foreground/background modelling approach (e.g., background subtraction) to detect the boats is ineffective. A possible solution consists in developing a classification-based detector, since it can operate on still images, thus avoiding the need of creating and updating a model of the background. To build the classifier it is necessary to learn a model from a set of labelled data instances (training phase). Then, in the testing phase, the learned model can be exploited to classify a test instance [19]. Different methods can be used to create the classifier (e.g., [6], [20]). The Haar-like features based approach described in [21] works in real-time, differently from other existing methods (e.g., [22], [23]), which are more computationally expensive. In particular, three different types of Haar-like features are used: Edge, line, and center-surround features (see Fig. 3a).

It is worth noticing that the Haar classifier has been originally designed for face detection, thus we have suitably adapted it for detecting boats. In particular:

- a 60×30 rectangular window is used, instead of a squared one
- median filtering is applied to smooth the image in input to reduce the presence of wakes
- the thresholds of the Canny edge detector are determined to focus on long edges. In particular, in our OpenCV implementation, we have empirically set the two threshold values to $min = 50$ and $max = 70$ for all the images

We have decided to create a unique binary classifier, training it with samples containing multiple types of vessels captured from different view angles. To this end, we have exploited the OpenCV function `opencv_traincascade` that can train a cascade of boosted classifiers from a set of samples. The used sample data set[2] contains 1549 positive images (taken mostly from the Internet) and 4000 negative images (without boats). In particular, the positive set contains bulkers (247 samples), cruise ships (85), ferry boats (87), fishing vessels (253), naval ships (71), sailing boats (169), yachts (217), and small boats (420). The training stage has been stopped when the false alarm rate reached 5×10^{-6}, obtaining a 24-level binary classifier capable of detecting boats of different size,

[2]Positive and negative images can be found at `http://goo.gl/TuUpkg`

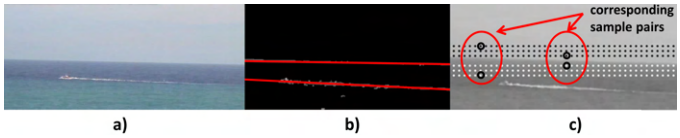


Fig. 4. Horizon line detection. a) Source image. b) Extracted Hough lines. c) Sample points used for validating the line (two corresponding pairs are highlighted).

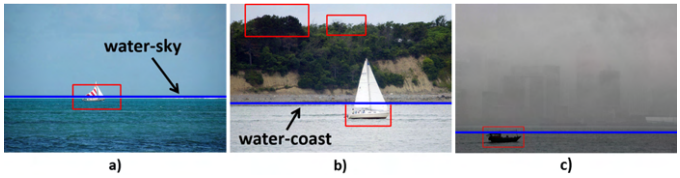


Fig. 5. a) Water-sky line. b) Water-coast line: The rich texture of the coastal background increases the probability of finding false positives. c) Horizon line detection in presence of fog. Pictures a and b are from the PASCAL VOC data set, c from the Internet.

with blurring noise, in presence of boat wakes and reflections on the water surface (see the examples in Fig. 3).

Horizon line detector. Along with the boats, it can be very useful to detect also the limit of the sea surface to discard false positives. The image in input (Fig. 4a) is processed by extracting the edges and then applying the Hough transform to the edge map, thus creating a list of candidate lines (Fig. 4b). Each candidate line is validated with respect to a set of sample points belonging to a rectangular region above and under it (see Fig. 4c). The grayscale intensity values of the points above the candidate line are compared with the values of the corresponding points under the line. If more than 90% of the compared pairs present different intensity values, then the line is considered valid. In such a way, long wakes can be filtered out since corresponding pairs above and under wakes have similar intensity values. An example is shown in Fig. 4b, where the lower line is discarded, while the upper line is considered a valid horizon line, since there is a considerable difference between the intensity values of the points over and under it.

Depending upon the heading of the camera, the horizon line detector differentiates between the water-sky and the water-coast line (Fig. 5). Indeed, the probability of finding false positives increases in presence of the coast, due to the rich texture of the background; thus, it is convenient to filter out the detections laying well above the water-coast line (Fig. 5b). The approach is rather robust and can work also in presence of fog (Fig. 5c).

Noise filter. To filter out false positives caused by waves and boat wakes, an additional level for the classifier has been created by using a special negative set of 4000 images, containing wakes and other false positive detection examples generated by the original 24-level classifier (see Fig. 6).

Moreover, to avoid false observations due to reflections, a filtering is carried out by counting the number of SURF [25] key points present in each potential observation in the current image (bounding box). If the number of key points in the bounding box is negligible, then the observation is

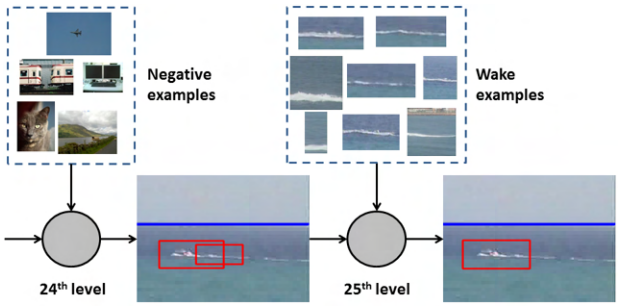


Fig. 6. An additional weak-classifier (25^{th} level) is used for wake removal. 4000 wake images are used as negative samples to train the new level.

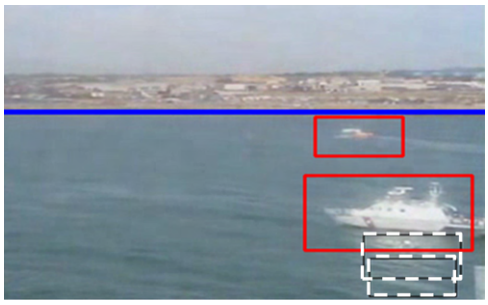


Fig. 7. False detections caused by reflections can be filtered out by analyzing the SURF key points. Dashed line bounding boxes are rejected detections.

discarded. Indeed, since SURF detection is not invariant to mirror reflections, the number of key points generated by reflections on the water surface is usually limited (an example is shown in Fig. 7).

IR data. The same classifier used for the EO images has been tested also with IR data. Differently from the approach adopted for the EO data, a pre-processing step has been performed for the IR images, in order to improve the results of the horizon-line detection. In particular, a normalization step is used to scale and to shift the source image (see Fig. 8). Since the classifier has been trained using only EO images, the fact that it can be successfully used on IR images supports the hypothesis that a Haar-like features based approach is suitable for boat detection. Moreover, the well-known uncalibrated polarity and halo issues in thermal imagery, which provoke problems in thresholding based method due to bright halo around dark objects and dark halo around bright objects, can be avoided by adopting a classifier based detection approach.

V. Tracking and Data Fusion

The approach described in this paper aims at enhancing the current VTS functions, generating a unique view where the information coming from the camera and the VTS tracks are merged together. We propose here a distributed tracking approach, able to fuse data from multiple heterogeneous and not synchronized sources. The input observations are provided by the above described Visual Detection module.

The problem of tracking multiple objects by using multiple sensors can be formulated as follows. Let $\mathcal{O} = \{o_1, \ldots, o_n\}$ be the set of all the moving objects, each one having a different identity, and $\mathcal{S} = \{s_1, \ldots, s_S\}$ be the set of all sensors,

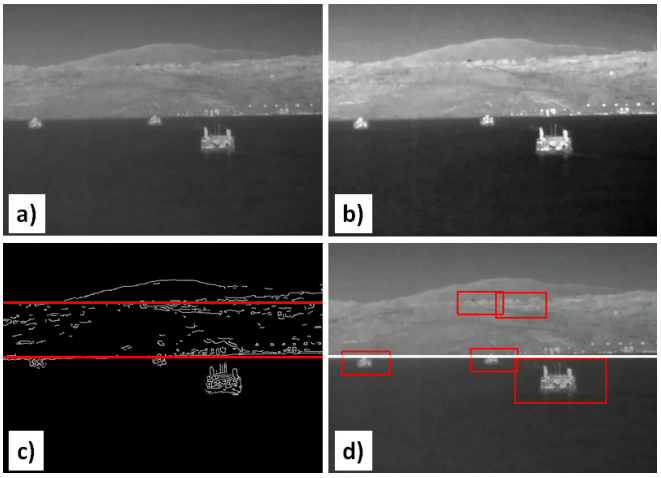


Fig. 8. IR detection. a) Source image. b) Normalized image. c) Candidate lines extraction. d) Detection with false positives over the the water-sky line discarded.

each one having an associated FOV, typically covering only a limited area of the scene. The total number n of objects that will be observed is unknown and the number l of the current objects in the scene, with $0 \leq l \leq n$, can change over time. The set of measurements (observations) about the objects in the FOV of a sensor $s \in \mathcal{S}$ at a time t is denoted by $z_{s,t} = \{z_{s,t}^{(1)}, \ldots, z_{s,t}^{(l)}\}$, where a measurement $z_{s,t}^{(i)}$ can be either an actual object or a false positive. The set of all the measurements gathered by all the sensors at time t is denoted by $\boldsymbol{z}_{\mathcal{S},t} = \{\boldsymbol{z}_{s,t} | s \in \mathcal{S}\}$. The history in time of all the measurements coming from the sensors is defined as $\boldsymbol{z}_{\mathcal{S},1:t} = \{\boldsymbol{z}_{\mathcal{S},j} : 1 \leq j \leq t\}$. Since the sensors can have different refresh rate, we do not assume that the measurements generated by the sensors are synchronized.

The goal is to determine an estimation of the positions $\boldsymbol{x}_{s,t} = \{x_{s,t}^{(1)}, \ldots, x_{s,t}^{(l)}\}$ for all the objects in the scene at time t in a distributed fashion, i.e., exploiting all the available sensors. In order to achieve this goal, a possible solution is to use the Bayesian recursion approach, defined as follows

$$p(\boldsymbol{x}_{s,t}|\boldsymbol{z}_{\mathcal{S},1:t}) = \frac{p(\boldsymbol{z}_{\mathcal{S},t}|\boldsymbol{x}_{s,t})p(\boldsymbol{x}_{s,t}|\boldsymbol{z}_{\mathcal{S},1:t-1})}{\int p(\boldsymbol{z}_{\mathcal{S},t}|\boldsymbol{x}_{s,t})p(\boldsymbol{x}_{s,t}|\boldsymbol{z}_{\mathcal{S},1:t-1})d\boldsymbol{x}_{s,t}} \quad (1)$$

$$p(\boldsymbol{x}_{s,t}|\boldsymbol{z}_{\mathcal{S},1:t-1}) = \int p(\boldsymbol{x}_{s,t}|\boldsymbol{x}_{s,t-1})p(\boldsymbol{x}_{s,t-1}|\boldsymbol{z}_{\mathcal{S},1:t-1})d\boldsymbol{x}_{s,t-1} \quad (2)$$

Eq. 1 and Eq. 2 represent a global recursive update that can be computed if and only if a complete knowledge about the scene is available. Since this is not the case, we approximate the above exact optimal Bayesian computation by means of a Distributed Particle Filter-based algorithm (see Fig. 9). In particular, we extend to a multi-sensor scenario the PTracking[3] method, which is an open-source tracking algorithm based on a Distributed Multi-Clustered Particle Filtering [17], [26], [27].

The estimation of the positions $\boldsymbol{x}_{s,t}$ is given by the vectors $\boldsymbol{I}_{s,t}, \boldsymbol{\Lambda}_{s,t}, \boldsymbol{M}_{s,t}, \boldsymbol{\Sigma}_{s,t}$ containing information about the identity ($\boldsymbol{I}$), the weight ($\boldsymbol{\Lambda}$), the mean ($\boldsymbol{M}$) and the standard deviation ($\boldsymbol{\Sigma}$) of each object, represented as a Gaussian Mixture

[3]PTracking can be found at `https://github.com/fabioprev`

Input: Measurements $z_{s,t}$, local track numbers $i_{s,t-1}$, global track numbers $I_{s,t-1}$

Data: Set of local particles $\tilde{\xi}_{s,t}$, set of global particles $\tilde{\xi}_{\mathcal{S}',t}$, local GMM set $\mathcal{L}$, global GMM set $\mathcal{G}$

Output: Global estimations $x_{s,t} = (\boldsymbol{I}_{s,t}\boldsymbol{\Lambda}_{s,t}, \boldsymbol{M}_{s,t}, \boldsymbol{\Sigma}_{s,t})$

// Local Estimation Layer
begin

$\tilde{\xi}_{s,t} \sim \pi_t(x_{s,t}|x_{s,t-1}, z_{s,t})$

Re-sample by using the SIR principle

$\mathcal{L} \leftarrow KClusterize(\tilde{\xi}_{s,t})$

$(\boldsymbol{i}_{s,t}, \boldsymbol{\lambda}_{s,t}, \boldsymbol{\mu}_{s,t}, \boldsymbol{\sigma}_{s,t}) \leftarrow DataAssociation(\mathcal{L}, i_{s,t-1})$

Communicate belief $(\boldsymbol{i}_{s,t}, \boldsymbol{\lambda}_{s,t}, \boldsymbol{\mu}_{s,t}, \boldsymbol{\sigma}_{s,t})$ to other sensors

end

// Global Estimation Layer
begin

Collect $\mathcal{L}_{S'}$ from a subset $\mathcal{S}' \subseteq \mathcal{S}$ of sensors within a Δt

$\tilde{\xi}_{\mathcal{S}',t} \sim \tilde{\pi} \leftarrow \sum_{s\in\mathcal{S}'} \boldsymbol{\lambda}_{s,t}\mathcal{N}(\boldsymbol{\mu}_{s,t}, \boldsymbol{\sigma}_{s,t})$

Re-sample by using the SIR principle

$\mathcal{G} \leftarrow KClusterize(\tilde{\xi}_{\mathcal{S}',t})$

$(\boldsymbol{I}_{s,t}, \boldsymbol{\Lambda}_{s,t}, \boldsymbol{M}_{s,t}, \boldsymbol{\Sigma}_{s,t}) \leftarrow DataAssociation(\mathcal{G}, I_{s,t-1})$

end

Fig. 9. PTracking algorithm. Each sensor runs this two-tired architecture to perform the tracking in the local and global reference frames.

Input: Particle set $\mathcal{P} = \{p_1, \ldots, p_m\}$

Data: Set of centroids $\mathcal{F}$, cluster of particles c_i, sets of Gaussian clusters $\mathcal{Q}$ and $\mathcal{C}$

Output: GMM set $(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma})$

$initialize\ \mathcal{F} = \varnothing$

for all $p_i \in \mathcal{P}$ **do**
  if $\forall f_k \in \mathcal{F}\ \{\|p_i, f_k\| > \delta_{model}\}$ **then**
    $\mathcal{F} \leftarrow \mathcal{F} \cup \{p_i\}$

$c_i = \varnothing \quad \forall i \in [1, |\mathcal{F}|]$

for all $p_i \in \mathcal{P}$ **do**
  for all $f_k \in \mathcal{F}$ **do**
    if $\|p_i, f_k\| < \delta_{model}$ **then**
      $c_k \leftarrow c_k \cup \{p_i\}$

$initialize\ \mathcal{C} = \varnothing$

for all c_i **do**
  if $c_i \not\sim \mathcal{N}(\mu, \sigma)$ **then**
    $\mathcal{Q} \leftarrow KClusterize(c_i)$
    for all $q_j \in \mathcal{Q}$ **do**
      if $q_j \sim \mathcal{N}(\mu, \sigma)$ **then**
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{q_j\}$

compute $(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ *from* $\mathcal{C}$

Fig. 10. KClusterize algorithm.

Model (GMM). The size of the vectors can vary during the execution of the tracking algorithm, depending on the number of detected objects.

The estimation process is made of three main steps: *(i)* the prediction step, which computes the evolution of the estimations $\boldsymbol{x}_{s,t}$ given the observations $\boldsymbol{z}_{s,t}$ provided by the sensors; *(ii)* the clustering step, which groups the estimations determining their GMMs parameters; *(iii)* the data association step, which assigns each observation to an existing track by considering the history of all the existing tracks.

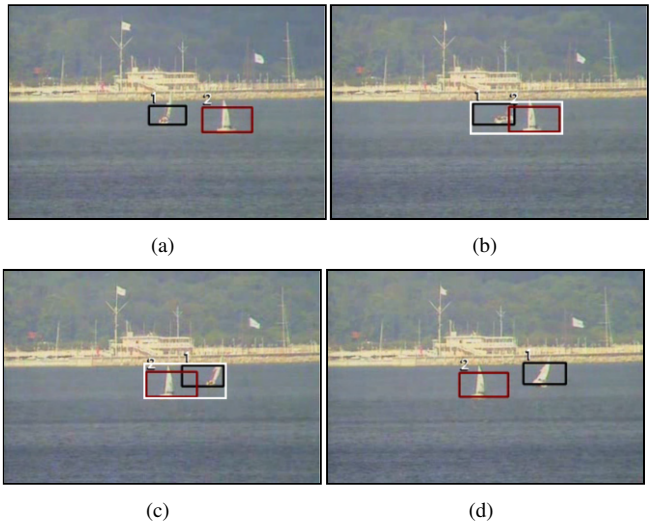
Prediction. The Particle Filter uses an initial guessed distribution, based on a *transition state* model. Then, using the previous state $\boldsymbol{x}_{s,t-1}$, the transition model, given by the measurements $\boldsymbol{z}_{s,t}$, is applied. From this guessed distribution, a set of samples is drawn and weighted exploiting the current observation $\boldsymbol{z}_{s,t}$. Finally, the *Sampling Importance Resampling* (*SIR*) [28] principle is used to re-sample the particles, which are then clustered to determine the parameters of the final GMM model.

Clustering. A novel clustering algorithm, called *KClusterize* (see Fig. 10), is used for the clustering phase. KClusterize is designed for fulfilling the following requirements: *(i)* the number of objects to detect is not known *a priori*; *(ii)* low computational load is needed for real-time applications; *(iii)* each cluster has a Gaussian distribution. First, the particles are grouped into clusters. Then, a validation step is applied to verify that each cluster actually represents a Gaussian distribution. All the non-Gaussian clusters are split (if possible) in Gaussian clusters. Finally, the obtained clusters form a GMM set $(\boldsymbol{\lambda}_{s,t}, \boldsymbol{\mu}_{s,t}, \boldsymbol{\sigma}_{s,t})$ representing the estimations performed by the sensor s at time t.

As a difference with other clustering methods (e.g., *k-means*, *EM*, *BSAS* or *QT-Clustering*), KClusterize does not require to know in advance the number of clusters, has a linear complexity, and all the obtained clusters reflect a Gaussian distribution.

Data association. An identity (i.e., a track number) has to be assigned to each object, by associating the new observations to the existing tracks. This is the crucial step for any tracking algorithm: The direction, the velocity, and the position of the objects are the features involved in the association algorithm (Fig. 12). We consider two moving tracked objects having the same direction if the angle between their trajectories is less than 10°.

The data association step is further complicated by complete and partial occlusions, which can occur when boats are aligned with respect to the camera view or when they are close to each other. Our solution is to consider the collapsing tracks as a group, instead of tracking them separately (see Fig. 11). When two or more tracks have their bounding boxes moving closer to each other (Fig. 11a), the tracker saves their color histograms and starts considering them as a group (Fig. 11b and Fig. 11c) — the histograms are used as models for re-identifying the objects when the occlusion phase is over (Fig. 11d). A group evolves taking into account both the estimated trajectory and the observations coming from the detector. When an occluded object becomes visible again, the stored histograms are used



(a) (b) (c) (d)

Fig. 11. Group tracking. Occlusions are handled by considering the collapsing tracks to form a group instead of tracking them separately.

Input: GMM set $(\boldsymbol{\Lambda}_{s,t}, \boldsymbol{M}_{s,t}, \boldsymbol{\Sigma}_{s,t})$, global track numbers $I_{\mathcal{S}',t-1}$

Data: e_s estimation performed by the sensor s, set of estimations to fuse $\mathcal{A}$

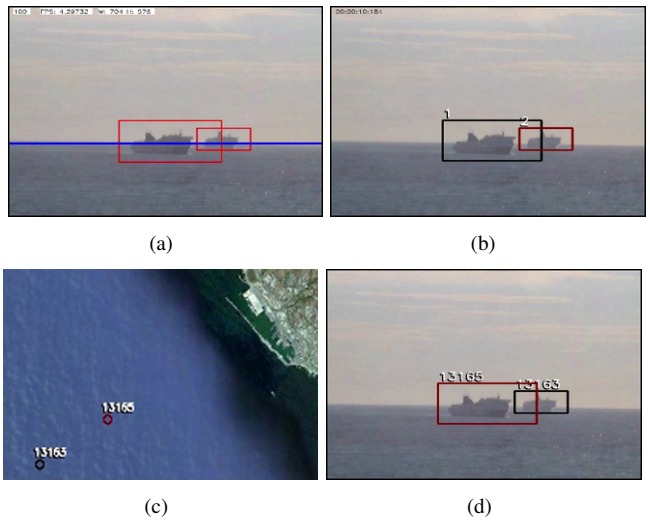
Output: Global estimations $x_{s,t} = (\boldsymbol{I}_{s,t}, \boldsymbol{\Lambda}_{s,t}, \boldsymbol{M}_{s,t}, \boldsymbol{\Sigma}_{s,t})$

for all $s \in \mathcal{S}' \subseteq \mathcal{S}$ **do**
  $\mathcal{A} = \varnothing$
  for all $e_s \in \boldsymbol{M}_{s,t}$ **do**
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{e_s\}$
    for all $\tilde{s} \in \mathcal{S}' \subseteq \mathcal{S}, s \neq \tilde{s}$ **do**
      for all $e_{\tilde{s}} \in \boldsymbol{M}_{\tilde{s},t}$ **do**
        if sameDirection$(e_s, e_{\tilde{s}})$ **and** sameModule$(e_s, e_{\tilde{s}})$ **and** close$(e_s, e_{\tilde{s}})$ **then**
          $\mathcal{A} \leftarrow \mathcal{A} \cup \{e_{\tilde{s}}\}$
  $I_{\mathcal{A}} \leftarrow Re\text{-}Identification(\mathcal{A})$
  // Re-Identification has failed, we assign a new track number
  if $I_{\mathcal{A}}$ is invalid **then**
    $I_{\mathcal{A}} \leftarrow maxTrackNumber + 1$
  $x_{s,t} \leftarrow x_{s,t-1} \cup FuseData(\mathcal{A}, \boldsymbol{\Lambda}_{s,t}, \boldsymbol{\Sigma}_{s,t}, I_{\mathcal{A}})$

Fig. 12. Data Association algorithm.

to re-assign the correct identification number, belonging to the corresponding previously registered track.

Data fusion. As discussed above, the information coming from the camera and the VTS system are fused in order to generate an enhanced and reliable believe-state of the tracked boats (see figures 13, 14, and 15). The Data Fusion stage is complicated by the lack of a common reference frame: Indeed, the calibration parameters of the camera are often not available. To cope with this problem, we devised the following algorithm. Let v_i^C be the velocity vector of the boat i in the camera reference frame C, and $\mathcal{V}^R$ be the set of velocity vectors of boats in the VTS reference frame R. The best matching candidate in $\mathcal{V}^R$ to be fused with the boat i in



(a) (b) (c) (d)

Fig. 13. Data fusion. a) Output provided by the detection algorithm. b) Output of the PTracking algorithm for the camera sensor. c) Output of the PTracking algorithm for the VTS sensor. d) Final output: VTS information are provided into the visual frame.

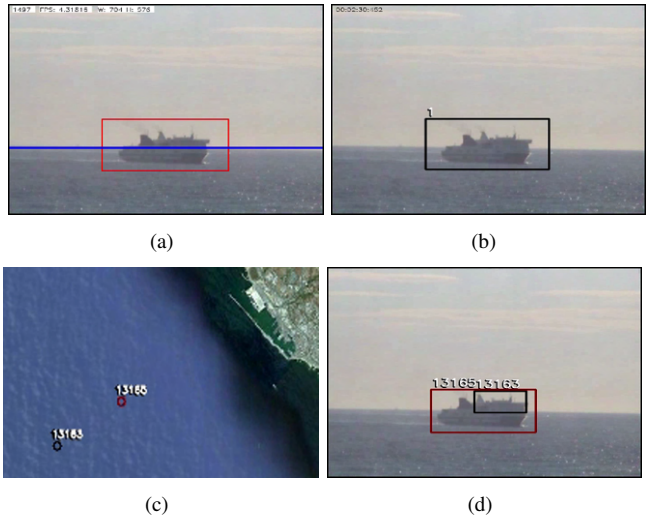
the camera reference frame is selected by computing for each $v_j^R \in \mathcal{V}^R$:

$$\begin{bmatrix} v_{x_j}^R \\ v_{y_j}^R \end{bmatrix}^T \cdot \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \simeq \begin{bmatrix} v_{x_i}^C \\ v_{y_i}^C \end{bmatrix}^T \tag{3}$$

where the rotation parameter θ between the camera and the VTS reference frames is calculated by means of a *Policy Gradient* algorithm [29]. More specifically, the optimization process finds the best value for θ starting from a predefined initial value (we use $\theta = 15°$ in our experiments). Afterwards, tracking and data fusion are performed by using the initial value assigned to θ as rotation parameter between the camera and the VTS frame to obtain quantitative results (see Section VI-C). Then, we perform two computations in parallel that use a lower and greater value of θ, respectively. When the computations are done, we recompute the quantitative results, checking for which direction the performance improve. We set this new value as "*initial value*" and the algorithm is executed until there is a significant variation in the quality metrics used to evaluate the performance of the system. This approach to calibrate the θ rotation parameter between the camera and the VTS data requires either a manually labeled ground-truth of a sequence of the input source or the help of a human user for updating the θ parameter. In the case of multiple matches for v_i^C, the Data Fusion algorithm continues to calculate the evolution for all the current matches, until a unique match is finally found.

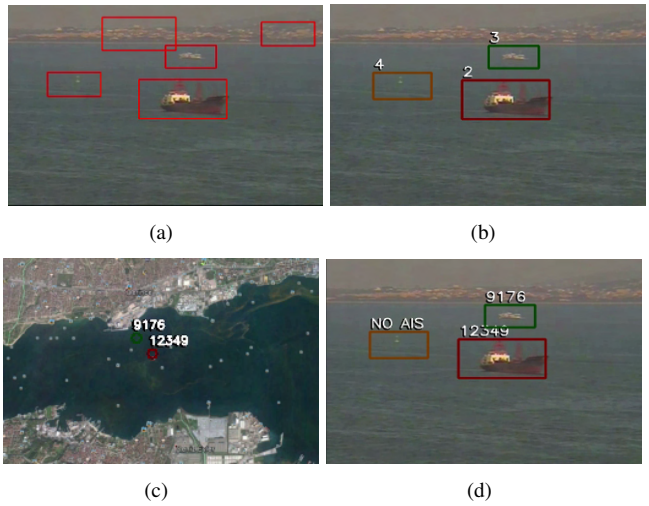
VI. Experimental Results

In order to quantitatively evaluate the performance of our approach, we have carried out experiments — using publicly available data set — on the main components of the architecture, i.e., the Visual Detection, the Visual Tracking, and the Data Fusion modules. In particular, we have quantitatively measured the detection accuracy, the tracking precision, the



(a) (b) (c) (d)

Fig. 14. Data fusion in the case of a total occlusion. Even if the visual tracking provides the estimation only for the boat in the foreground, the Data Fusion module can show also the boat in the background.



(a) (b) (c) (d)

Fig. 15. Data fusion in the case of a non-AIS boat. The visual tracker highlights the presence of a boat, which can be recognized by as human operator.

data fusion quality and the computational speed of the proposed solutions. Moreover, we have tested the performance of the whole system (i.e., tracking plus data fusion) when VTS data are available.

A. Data Set

All data sets used in this experimental evaluation can be found at the MarDCT Maritime Detection, Classification and Tracking [24] database, containing images and videos with ground-truth annotations. The videos have been recorded with varying observing angles and weather conditions. In particular, for each video details about the camera type (static or moving, EO or IR) and the location and time of day, as well as foreground masks to evaluate the image segmentation and ground-truth annotations with bounding box vertices and identification numbers to evaluate tracking results are provided. At this moment, the MarDCT data set contains:

1) EO and IR videos recorded in a VTS centre in Italy
2) EO and IR videos taken in a Northern Europe centre
3) EO videos from the ARGOS system [8] monitoring the Grand Canal in Venice, Italy
4) EO videos from a port in Eastern Asia

The MarDCT database also contains two examples where visual and VTS information are recorded together to allow data fusion tests.

B. Detection Results

The accuracy of the detection process has been evaluated by using both EO and IR images (from real VTS centres).

Detection metrics. The detection accuracy has been measured considering two different metrics: True Positive Rate (also known as Recall) and False Alarm Rate (also known as False Discovery Rate), defined as follows:

$$TPR = \frac{TP}{TP + FN} \quad (4) \qquad FAR = \frac{FP}{TP + FP} \quad (5)$$

where TP are the true positives, i.e., correctly detected boats, FN are the false negatives, i.e., not detected boats, and FP are the false positives, i.e., the number of background regions wrongly detected as objects of interest. The TPR measures the number of true detections on the total number of examples, while the FAR gives a measure of the false detections over all the detections generated by the system.

Detection results on EO data. A test data set for evaluating the detection on EO images has been created by extracting 50 frames (a frame every 5 seconds) from 9 different EO videos, for a total of 450 EO images. The videos are recorded with varying light conditions and camera positions in real VTS centres and contains boats with different shape and size. Table I shows the results. Since the videos are from different scenarios (using varying sensors and frame size), we have reported also the used detection size, which depends on the zoom level. A good performance of our approach can be observed and, in particular, a considerable reduction of the false alarm rate is achievable by activating the horizon line detection filter.

Detection results on IR data. The same classifier used for the EO images has been tested with IR data. A set of 150 uniformly selected images from 3 videos (50 frames per video, with each frame extracted every 5 seconds) recorded in a VTS centre in Northern Europe has been considered for measuring the detection accuracy with IR images. The results on the IR data, shown in Table II, demonstrate the effectiveness of the classification-based approach and the positive contribution of the horizon line filtering. Indeed, specially when dealing with IR images, noise sources from the coast strongly increase the number of false positives.

C. Visual Tracking Results

Three annotated video clips from three different real sites have been used to quantitatively evaluate the Visual Tracking results:

1) *occlusions-1* containing two ferry-boats with a complete occlusion.

TABLE I
VISUAL DETECTION RESULTS ON EO DATA.

Video	Horizon Detection	TPR	FAR	Real Size	Detection Size
occlusions-1	NO	0.725	0.275	704×576	704×576
	YES	**0.780**	**0.200**		
occlusions-2	NO	0.988	0.012	469×384	234×192
	YES	**1.000**	**0.000**		
occlusions-3	NO	0.993	0.007	469×384	234×192
	YES	**1.000**	**0.000**		
high-view	NO	0.664	0.336	853×480	234×192
	YES	**0.691**	**0.309**		
boats-1	NO	1.000	0.000	939×768	703×576
	YES	**1.000**	**0.000**		
boats-2	NO	0.821	0.179	469×384	352×288
	YES	**0.842**	**0.158**		
wakes-1	NO	0.825	0.175	426×320	319×240
	YES	**0.853**	**0.147**		
wakes-2	NO	0.763	0.237	938×768	703×576
	YES	**0.781**	**0.219**		
wakes-3	NO	0.939	0.061	938×768	703×576
	YES	**0.959**	**0.041**		

TABLE II
VISUAL DETECTION RESULTS ON IR DATA.

Video	Horizon Detection	TPR	FAR	Real Size	Detection Size
ir-1	NO	0.798	0.135	938×768	703×576
	YES	**0.865**	**0.130**		
ir-2	NO	0.583	0.388	938×768	469×384
	YES	**0.612**	**0.417**		
ir-3	NO	0.773	0.227	938 ×768	703×576
	YES	**0.791**	**0.209**		

2) *occlusions-2* showing two small sailing boats intersecting their trajectories with partial occlusion.
3) *high-view* in which boats are seen from an high view.

The first two videos have been chosen to demonstrate the ability of our tracker to deal with partial and complete occlusions as well as with missing observations. The third video has been used to demonstrate the robustness of our approach even in presence of a particular view, in which the boats are captured from an high view. The above described videos, along with their ground-truth data, can be downloaded from the MarDCT database.

Tracking metrics. We use the CLEAR MOT [30] metrics *MOTA* and *MOTP* together with *Precision* and *Recall* to measure quantitatively the performance of the tracking method. The *Multiple Object Tracking Accuracy* (*MOTA*) is defined as:

$$MOTA = 1 - \frac{\sum_{t=1}^{N_{frames}} (c_m(m_t) + c_f(fp_t) + cs(ID\text{-}S_t))}{\sum_{t=1}^{N_{frames}} N_G^{(t)}} \quad (6)$$

where, after computing the mapping for frame t, m_t is the number of misses, fp_t is the number of false positives, $ID\text{-}S_t$ is the number of ID mismatches in frame t considering the mapping in frame $(t-1)$ and $N_G^{(t)}$ is the number of ground-truth objects in the t-th frame. The values used for the weighting functions in this evaluation were $c_m = c_f = 1$ and $c_s = log_{10}$.

TABLE III
QUANTITATIVE RESULTS FOR THE VISUAL TRACKING MODULE.

Video	Horizon Detection	MOTA	MOTP	Precision	Recall	False Positives
occlusions-1	NO	0.808	0.607	0.997	0.814	0.005
	YES	**0.815**	**0.613**	**1.000**	**0.815**	**0.000**
occlusion-2	NO	0.905	0.542	0.952	0.939	0.062
	YES	**0.910**	**0.554**	**0.955**	**0.955**	**0.045**
high-view	NO	0.901	0.601	0.972	0.923	0.031
	YES	**0.910**	**0.604**	**0.982**	**0.927**	**0.017**

The *Multiple Object Tracking Precision* (*MOTP*) is defined as:

$$MOTP = \frac{\sum_{i=1}^{N_{mapped}} \sum_{t=1}^{N_{frames}^{(t)}} \left[\frac{|G_i^{(t)} \cap D_i^{(t)}|}{|G_i^{(t)} \cup D_i^{(t)}|} \right]}{\sum_{t=1}^{N_{frames}} N_{mapped}^{(t)}} \quad (7)$$

where N_{mapped} refers to the mapped system output objects over an entire reference track taking into account splits and merges. $N_{mapped}^{(t)}$ refers to the number of mapped objects in the $t\text{-}th$ frame, $G_i^{(t)}$ denotes the i-th ground-truth object in the t-frame and $D_i^{(t)}$ denotes the tracked object for $G_i^{(t)}$.

Recall is defined as in Eq. 4, while Precision is computed as follows:

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

To obtain the precision score, we calculated the spatio-temporal overlap between the reference tracks and the system output tracks.

Tracking quantitative results. Table III shows the quantitative results for the visual tracking module on three different videos, which contain partial and total occlusions[4]. For all the considered videos, tracking results are very good: This is proved by the high values of MOTA and MOTP metrics. Moreover, there are also few false positives thanks to the proper temporal filtering performed by the PTracking method. Complete (contained in *occlusion-1* video) and partial (in *occlusion-2* video) occlusions are correctly handled by the tracking algorithm, thanks to the use of the future object motion prediction for re-identifying previously occluded objects.

D. Data Fusion Results

We used the *occlusion-1* video as benchmark for measuring the performance of the whole pipeline (tracking plus data fusion), since VTS data are available for it along with visual information. In addition, we analyzed a video called *three-boats* containing two large vessels equipped with AIS and a small boat without AIS.

[4]The videos showing the results of the visual tracking approach can be downloaded from: `http://goo.gl/uVYm4T`, `http://goo.gl/efRXVr`, `http://goo.gl/ySuAVM`

TABLE IV
QUANTITATIVE RESULTS FOR THE DATA FUSION MODULE.

Video	Horizon Detection	MOTA	MOTP	Precision	Recall	False Positives
occlusions-1	NO	0.969	0.863	1.000	0.947	0.000
	YES	**0.975**	**0.871**	**1.000**	**0.950**	**0.000**
three-boats	NO	0.916	0.843	0.989	0.922	0.000
	YES	**0.921**	**0.845**	**0.995**	**0.925**	**0.000**

TABLE V
COMPUTATIONAL SPEED FOR THE COMPLETE PIPELINE.

Frame Size	FPS (2 cores)	FPS (8 cores)
319 × 261	15.5	81.3
352 × 288	13.8	75.6
414 × 338	10.1	51.7
586 × 479	6.2	29.4
704 × 576	4.5	23.5

To evaluate the Data Fusion algorithm performance, we considered the CLEAR MOT metrics MOTA and MOTP together with Precision and Recall indexes, both activating and not activating the horizon line filter. The horizon detection can help in slightly improving the result, since the observations generated during the detection phase are better. Table IV shows the obtained results[5]. What is important to note is that the tracking performance is significantly improved when multiple data sources (in such a case camera and VTS) are used. Indeed, by comparing Table III and Table IV, the MOTP raises from 0.613 to 0.871 and the MOTA from 0.815 to 0.975.

E. Runtime Performance

We have measured the runtime performance of the complete system. The computational speed in terms of frames per second (FPS) has been measured on live data coming from a EO/IR camera in a real site, using an Intel Core 2 U7300 1.30 GHz, 4 GB RAM (2 cores) and an Intel Core i7 3770 3.40 GHz, 16 GB RAM (8 cores). The results are shown in Table V demonstrating that the complete approach is scalable and that the runtime performance increases when more computational power is available.

F. Discussion

The analysis of the experimental results leads to the following considerations. The Haar-like features based approach is an effective solution for boat detection with moving and zooming cameras in the maritime domain. A high detection rate is obtained on real data, both EO and IR images. It is worth noting that our detector can deal with low quality, compressed images coming from real sites, where captured boats navigate far from the coast. A detection approach based on a Haar classifier inherently produces a high true positive rate, but at the price of having an elevated false alarm rate. The horizon line detection filter is crucial in lowering the FAR, allowing for an improvement of the overall detection performance.

Moreover, the observations generated by the boat detector are sent to the Visual Tracking module that further reduces the false detections thanks to its temporal filtering. This is demonstrated by the high MOTA values as well as by the low percentage of false positives obtained in the experiments on tracking (see Table III). MOTP results are related to the quality of the observations in input, which is influenced by the low detection rate caused by occlusions. Finally, the approach is suitable for a real-time application. With a commercial CPU it is possible to achieve a real-time (29 FPS) processing speed for 586×479 images.

[5]The two videos showing the results of the proposed data fusion approach can be downloaded from `http://goo.gl/7TmXTr` and `http://goo.gl/Akl90H`

VII. CONCLUSIONS

In this paper, a modular architecture for improving automatic maritime surveillance systems with visual information is presented. The key idea is to use the camera as the main sensor, differently from the traditional VTS systems that use radars. Replacing radar sensors with cameras allows for deploying the system in populated areas at a lower cost. A major advantage of the proposed approach is the possibility of providing a global view of the captured scene, by adding a visual dimension to radar and AIS data, which is very effective for the user.

A quantitative experimental evaluation has been conducted by considering a publicly available large data set, containing images and videos from real working VTS sites. The accuracy in both the detection and tracking phases has been analysed, showing the effectiveness of the proposed methods in detecting and tracking boats, while maintaining real-time performance. Furthermore, the complete pipeline (i.e., visual tracking plus VTS data fusion) has been tested, demonstrating that the proposed solution is feasible for enhancing the capability of existing VTS systems.

As future work, we intend to perform a deep analysis on the scalability of the proposed approach in terms of the number of objects detected and tracked. Moreover, we will investigate possible improvements deriving from the inclusion of soft data from human operators.

ACKNOWLEDGMENT

The authors would like to thank Selex ES S.p.A - A Finmeccanica Company for the material and the support provided for this work as well as Miss. Federica Fieno for her invaluable help in the ground-truth labeling.

REFERENCES

[1] World Ocean Review, "Global shipping - a dynamic market." [Online]. Available: worldoceanreview.com/en/wor-1/transport/global-shipping
[2] J. García, J. L. Guerrero, A. Luis, and J. M. Molina, "Robust sensor fusion in real maritime surveillance scenarios," in *International Conference on Information Fusion*, 2010, pp. 1–8.
[3] G. Saur, S. Estable, K. Zielinski, S. Knabe, M. Teutsch, and M. Gabel, "Detection and classification of man-made offshore objects in terrasar-x and rapideye imagery: selected results of the demarine deko project," in *Spain OCEANS*, 2011, pp. 1–10.

[4] W. Kruger and Z. Orlov, "Robust layer-based boat detection and multi-target-tracking in maritime environments," in *International Waterside Security Conference*, 2010, pp. 1–7.
[5] S. Fefilatyev, D. Goldgof, M. Shreve, and C. Lembke, "Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system," *Ocean Engineering*, vol. 54, pp. 1–12, 2012.
[6] M. D. R. Sullivan and M. Shah, "Visual surveillance in maritime port facilities," in *SPIE Defense and Security Symposium*, 2008, pp. 697 811–697 811.
[7] H. Wei, H. Nguyen, P. Ramu, C. Raju, X. Liu, and J. Yadegar, "Automated intelligent video surveillance system for ships," in *SPIE Defense, Security, and Sensing*, vol. 7306, 2009, p. 73061N.
[8] D. D. Bloisi and L. Iocchi, "Argos - a video surveillance system for boat traffic monitoring in Venice," *Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 07, pp. 1477–1502, 2009.
[9] J. Leggat, T. Litvak, I. Parker, A. Sinha, S. Vidalis, and A. Wong, "Study on persistent monitoring of maritime, Great Lakes and St. Lawrence seaway border regions," DTIC Document, Tech. Rep., 2011.
[10] D. D. Bloisi, L. Iocchi, D. Nardi, and M. Fiorini, "Integrated visual information for maritime surveillance," in *Clean Mobility and Intelligent Transport Systems*, 2015, pp. 237–264.
[11] I. Bechar, F. Bouchara, T. Lelore, V. Guis, and M. Grimaldi, "Uncertainty fusion based object recognition and tracking in maritime scenes using spatiotemporal active contours," in *International Conference on Computer Vision Theory and Applications*, vol. 1, 2014, pp. 682–689.
[12] S. Fefilatyev, D. Goldgof, and C. Lembke, "Tracking ships from fast moving camera through image registration," in *International Conference on Pattern Recognition*, 2010, pp. 3500–3503.
[13] F. Bousetouane and B. Morris, *Off-the-Shelf CNN Features for Fine-Grained Classification of Vessels in a Maritime Environment*. Springer, 2015, pp. 379–388.
[14] B. Qi, T. Wu, B. Dai, and H. He, "Fast detection of small infrared objects in maritime scenes using local minimum patterns," in *International Conference on Image Processing*, 2011, pp. 3553–3556.
[15] A. L. Bustamante, J. M. Molina, and M. A. Patricio, "Information fusion as input source for improving multi-agent system autonomous decision-making in maritime surveillance scenarios," in *International Conference on Information Fusion*, 2014, pp. 1–8.
[16] E. Marti, A. Luis, J. Garcia, S. Onate, C. Sanchez, and S. Gonzalez, "Fusion of sensor data and intelligence in FITS," in *International Conference on Information Fusion*, 2013, pp. 342–349.
[17] F. Previtali and L. Iocchi, "PTracking: distributed multi-agent multi-object tracking through multi-clustered particle filtering," in *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2015, pp. 110–115.
[18] D. D. Bloisi, L. Iocchi, M. Fiorini, and G. Graziano, "Automatic maritime surveillance with visual target detection," in *International Workshop on Defense and Homeland Security Simulation*, 2011, pp. 141–145.
[19] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Computing Surveys*, vol. 41, no. 3, p. 15, 2009.
[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
[21] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.
[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 886–893.
[23] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1713–1727, 2008.
[24] D. D. Bloisi, L. Iocchi, A. Pennisi, and L. Tombolini, "ARGOS - Venice boat classification," in *Workshop on Vehicle Retrieval in Surveillance*, 2015, pp. 1–6.
[25] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: speeded up robust features," in *European Conference on Computer Vision*, 2006, pp. 404–417.
[26] A. Bordallo, F. Previtali, N. Nardelli, and S. Ramamoorthy, "Counterfactual reasoning about intent for interactive navigation in dynamic environments," in *International Conference on Intelligent Robots and Systems*, 2015, pp. 2943–2950.
[27] F. Previtali, G. Gemignani, L. Iocchi, and D. Nardi, "Disambiguating localization symmetry through a multi-clustered particle filtering," in *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2015, pp. 283–288.
[28] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
[29] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation." in *International Conference on Neural Information Processing Systems*, vol. 99, 1999, pp. 1057–1063.
[30] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 319–336, 2009.

Domenico D. Bloisi is research associate with Sapienza University of Rome, Italy. He received his PhD, M.Sc. and B.Sc. degrees in Computer Engineering from Sapienza University of Rome in 2010, 2006 and 2004, respectively. His main research interests are related to intelligent surveillance (including object detection, visual tracking and multiple sensor data fusion) and robotics.

Fabio Previtali is Ph.D. student at Sapienza University of Rome, Italy. He received his M.Sc. in Computer Engineering from Sapienza University of Rome in 2011 and his B.Sc. in Computer Engineering from the same University in 2009. His main research interests are related to life-long visual learning, distributed multi-object tracking, video surveillance and robotics.

Andrea Pennisi is post-doctoral researcher with Vrije Universiteit Brussel (VUB), Brussels. He received his Ph.D. and M.Sc. in Computer Engineering from Sapienza University of Rome in 2015 and 2010, respectively. He obtained his B.Sc. in Computer Engineering from University of Catania, Italy in 2007. His main research interests are related to image segmentation, multi-sensor surveillance, and crowd analysis.

Daniele Nardi is full professor with Sapienza University of Rome, Italy. His current research interests are mainly in the field of Artificial Intelligence in the area of Knowledge Representation and Reasoning and Multi Agent and Multi-robot systems. He is currently President of RoboCup and was co-chair of IEEE Technical Committee of International Workshop on Safety, Security and Rescue Robotics.

Michele Fiorini is principal engineer at Selex ES, a Finmeccanica Company, in Rome Italy where he is actually technical consortium leader for the ZSRN, a national system of radar control for maritime areas of Poland realized by Selex ES. His currently research interests are mainly on Vessel Traffic Services, Coastal Surveillance Systems, Intelligent Transport Systems and e-Navigation. Dr Fiorini is FIET, SMIEEE, Chartered UK Engineer (CEng) and Chairman of the IET Italy Network.