This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2775524, IEEE Transactions on Circuits and Systems for Video Technology

A Dataset for Airborne Maritime Surveillance Environments

Ricardo Ribeiro, *Member, IEEE,* Gonçalo Cruz, Jorge Matos, *Student, IST,* and Alexandre Bernardino, *Member, IEEE,*

***Abstract*—This work presents a dataset with surveillance imagery over the sea captured by a small size UAV. This dataset presents object examples ranging from cargo ships, small boats, life rafts to hydrocarbon slick. The video sequences were captured using different types of cameras, at different heights and different perspectives. The dataset also contains thousands of labels with positions of objects of interest. This was only possible to achieve with the labeling tool also described in this work. Additionally, using standard evaluation frameworks, we establish a baseline of results using algorithms developed by the authors which are better adapted to the maritime environment.**

***Index Terms*—Image databases, Hyperspectral imaging, Video surveillance,**

I. Introduction

MARITIME surveillance is a key activity for many countries. It is important to assure the safe and secure use of the oceans for transportation and trade. It allows the control of fisheries to guarantee the protection of resources and ecosystems. Maritime surveillance also ensures that environmental regulations are applied, preventing oil spill and bilge dumping that have a severe impact on fauna, flora and also coast human populations. Despite being an important activity, to this day it is still a difficult endeavor. It implies the use of vessels, aircraft and satellites, usually in a complementary fashion. All these platforms have their own limitations and therefore there is demand for additional technologies.

In the last decade, Unmanned Aerial Vehicles (UAVs) have seen a huge increase not only in its deployment but also in its capabilities. Right now, UAVs offer promising technologies to aid remote sensing and oceanic surveillance. While traditional aircraft are equipped with heavy radars, UAVs normally have only light weight passive electro-optical sensors. Whereas in traditional aircraft, the crew analyses the data being gathered, in unmanned aircraft the system needs additional intelligence. The additional intelligence is used to replace the human on board or at least help the human operator on the ground.

Several methods have been developed to increase the processing capabilities, following the developments in other areas of computer vision and pattern recognition. In this area, many authors use their own video sequences which are not publicly available and do not allow any kind of comparison. More recently, deep learning had a significant impact on computer vision, in tasks like classification and detection. One of the factors that has been pointed out to its success is the current abundance of images available online, allowing the training of different learning algorithms. Yet, in the majority of datasets, like ImageNet [1] represented in Fig. 1(a), the objects of interest are dominant, *i.e.* occupy a significant area of the image. This is not the case in aerial surveillance where, most of the times, the object of interest is quite small. Additionally, aerial surveillance images are captured from a different point of view and have its own challenges. The perspective of objects may change during its observation. Phenomena that generally are not present on online images like severe glare, white caps caused by waves and boat wakes also introduce noise.

To answer the lack of specialized datasets is common that authors build their own datasets, for example as Figueira *et al.* [2] and Nambiar *et al.* [3]. Bloisi *et al.* [4] have presented MARDCT, a dataset used for detection, classification and tracking of boats using visible and Infrared (IR) cameras. In this case, images are captured mostly from buildings near congested marine routes. Objects of interest have a relatively big size when comparing to the image size. Moreover, because cameras are still, the background is static. A similar approach was followed in VAIS dataset [5], with visible and IR images being captured by cameras installed in a pier. While this might be useful for some surveillance applications, it is limited for the case of airborne surveillance. Recently, datasets with nadir oriented aerial images like VEDAI [6] and DLR 3K Munich [7] have been introduced. These are quite useful for the survey of an area but because of the necessary perspective corrections are not suited for airborne real time surveillance tasks. In that kind of task, there are movements of the aircraft that are transmitted to the camera and cause the background to change drastically. Moreover, the perspective is also very different from the shown in the previous datasets. Patino *et al.* [8] and Prasad *et al.* [9] have presented more challenging datasets, with cameras mounted on ships that cause background variations, although, the perspective in images like presented in Fig. 1(b) is still quite different from aerial images' case. As shown in Fig. 1(c), there are challenges like the small scale, glare and movement of background that are characteristic of airborne images.

Given that most images online do not answer our requirements, in this work we present a significant dataset [10] to enable more reproducible and comparable research for aerial surveillance in maritime surveillance scenarios. To our knowledge, this is the first publicly available dataset of video sequences in a maritime surveillance scenario,

R. Ribeiro, J. Matos and A. Bernardino are with the Institute for Systems and Robotics, Department of Electrical and Computer Engineering, Instituto Superior Técnico, 1049-001 Lisboa, Portugal

G. Cruz is with the Portuguese Air Force Research Center, 2715-021 Sintra, Portugal.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2775524, IEEE Transactions on Circuits and Systems for Video Technology

(a) (b) (c)

Fig. 1. Examples of typical images of boats for (a) Imagetnet [1], (b) VIS [9], and (c) our dataset [10].

captured by a small unmanned aircraft. The dataset is available to the scientific community upon request, see http://vislab.isr.tecnico.ulisboa.pt/seagull-dataset/ for instructions and additional example images. After access is granted, the videos sequences (and ground truth labels) can be easily browsed, previewed or downloaded.

A. Contributions

The main contribution of this paper to the community is the introduction of a public dataset of thousands of images captured from an airborne platform in maritime surveillance scenarios in very diverse conditions, and the evaluation of the performance of the state-of-the-art algorithms on typical maritime scenarios.

Typically, authors that suggest techniques in this context use their own private datasets. This, therefore, hampers the reproducibility of the methods and the comparison with new ones. Additionally, collecting and labeling these datasets is a laborious task, that might be very difficult and/or unsought amount of work for some authors. Consequently, this might preclude some groups to work on some of these techniques. To mitigate this problem, we propose a new tool to facilitate the labeling task and make this database available to be used by the scientific community.

Another very important factor is that many of the current state-of-the-art methods for detection and tracking are based on machine learning techniques, which require large amounts of data for the training stage. Our dataset not only presents a considerable amount of samples but those samples are representative of challenging situations, close to real world scenarios, with a strong presence of glare, wave crests, wakes, variation of perspective, and with objects of interest of different types, scales and shapes. Some of these characteristics are presented in Table I, in the figures throughout the paper, in particular Fig. 2, and on the database webpage (http://vislab.isr.tecnico.ulisboa.pt/seagull-dataset/), to which we add the following:

- more than 150000 images were captured;
- objects of interest were labeled by a human operator in thousands of images;
- images were captured using different types of sensors, in particular visible light, Long Wave Infra-red (LWIR) and hyperspectral sensors.
- different types of objects of interest were observed, namely cargo ships, smaller boats (27 meters long), sailing yachts, life rafts, dinghies and a hydrocarbon slick.

We also present some results from state of the art algorithms and from our own maritime oriented algorithms to serve as a baseline against which the scientific community can compare their own developed methods.

Given the aforementioned considerations, the main advantage of this work is the introduction of a dataset that enables authors to work with a freely available and annotated dataset, without the need of collecting and labeling the images themselves. Also, this work provides a new labeling tool and an evaluation baseline using standard metrics against which new methods can be compared.

B. Outline of the paper

This work is organized as follows: in Section II, we describe the aircraft and cameras used to capture the dataset as well as the scenarios that were considered; Section III presents the most relevant data about the sequences and the labeling process; Section IV contains information about the evaluation process. The results for detection and tracking are presented in Section V and finally, Section VI concludes the present work.

II. Acquisition setup

A. System architecture

1) Aircraft: The aerial platform used to build these sequences was an Alfa Extended, a UAV designed, built and operated by the Portuguese Air Force Research Center for research purposes, depicted in Fig. 3. It has a wingspan of 3.5 meters and a maximum take-off weight of 25 kg. This UAV can carry up to 10 Kg in payload and has an endurance of 8 hours. The propulsion is supplied by a gas engine, which is also connected to a generator that provides electrical power to all onboard systems. We use a commercial off-the-shelf autopilot (Piccolo II) that takes care of the low-level control, has a GPS receiver and internal sensors to determine its position, orientation and air speed. The autopilot also uses an additional Differential GPS module that increases navigation precision and allows automatic landings. To communicate with on board devices, the autopilot uses several serial ports and a UHF data-link to communicate with a ground control station.

2) Cameras: While designing the UAV, one of the priorities was the use of electro-optical sensors and therefore the aircraft has a dedicated bay and an unobstructed view. This allowed us to use both fixed and steerable cameras.

Because we wanted the dataset to be diverse, *i.e.*, to contain different operation conditions and different objects, we have also used cameras with different characteristics. The major distinction between cameras was their spectrum, in particular: we have used two cameras that operate solely in the visible spectrum, one LWIR camera, one camera with one CCD receiving visible spectrum radiation and another CCD capturing Near Infrared (NIR) and finally one hyperspectral camera sensitive to radiation in the NIR and visible spectrum.

Only one of the used cameras was steerable, all rest of cameras were rigidly mounted on the aircraft. Considering the fixed cameras and with the exception of the hyperspectral, they were mounted on the airframe and were pointing $90°$ left of the aircraft and approximately $45°$ from the horizontal

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2775524, IEEE Transactions on Circuits and Systems for Video Technology

Fig. 2. Example images of the sequences in Table I: (a) *seq01*, (b) *seq02*, (c) *seq03*, (d) *seq04*, (e) *seq05*, (f) *seq06*, (g) *seq07*, (h) *seq08*, (i) *seq09*, (j) *seq10*, (k) *seq11*, (l) *seq12*, (m) *seq13*, (n) *seq14*, (o) *seq15*, (p) *seq16*, (q) *seq17*, (r) *seq18* and (s) *seq19*. The images show a diverse set of examples of the difficulties particular to the maritime environments, namely: Sun reflections (b), (e), (f), (g), (i), (j), (o), (q), and (r); Wakes (b), (h), (o), (p), and (q); Multiple boats (a) (b), (d), (e), (f); Different visual aspect of the boat after rotation - compare (o) with (p); Scale variations - compare (m), (n) with (o), (p); Illumination variations - compare (a) with (b) or (o) with (p) ;

Fig. 3. Alfa Extended, a small UAV designed and manufactured at the Portuguese Air Force Research Center.

plane, as represented in Fig. 4. The hyperspectral was pointing downwards, with its optical axis aligned with the vertical, as shown in Fig. 5. The steerable camera was mounted on the bottom of the aircraft and had the ability to pan, tilt and zoom.

As stated previously, we have used two visible spectrum cameras. The simplest was a GoPro Hero 2. It has a $1/2.5$" visible light CMOS sensor with a field of view of 170° and captured video with a resolution of 1920×1080 pixels. This camera works independently of the onboard systems, which makes it quite flexible to use but its images can only be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2775524, IEEE Transactions on Circuits and Systems for Video Technology

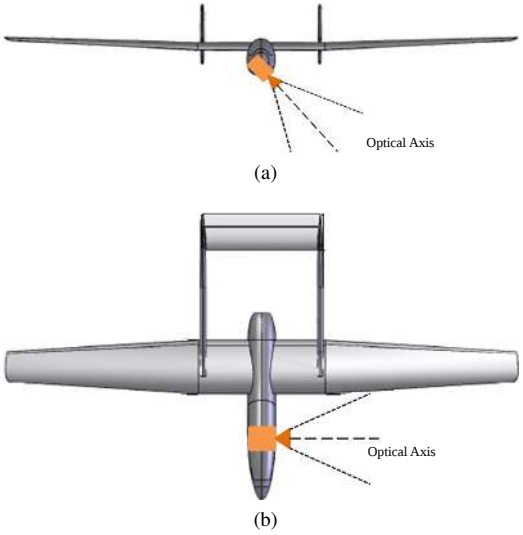


Fig. 4. Front (a) and top (b) views of the orientation of fixed cameras (with the exception of the hyperspectral camera).

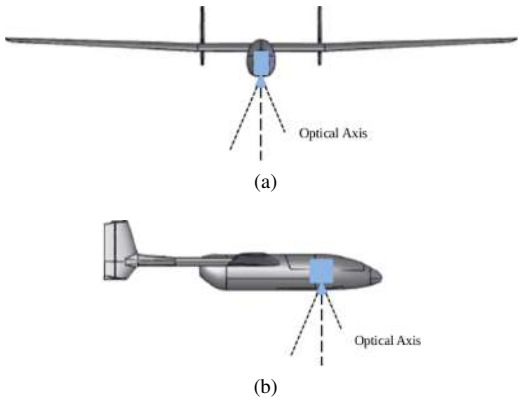


Fig. 5. Front (a) and side (b) views of the orientation of the hyperspectral camera.

processed off-line. The other visible spectrum camera was Tase 150, equipped with a $1/4$" CCD sensor that captured NTSC video, typically with a horizontal field of view of $42.2°$. This camera was mounted on a structure that was controlled by the autopilot and its analog video output was acquired by the onboard computer.

The LWIR camera that was installed was a GOBI 384. This camera is sensitive to radiation with wavelengths from 8 to 14 μm, which corresponds to most of thermal radiation emitted by bodies at ambient temperature. This camera has an ethernet interface that is connected directly to the airborne systems. The other camera with that kind of interface is JAI AD-080. This camera has two CCD, one that receives visible light and another that receives NIR. Both sensors have a resolution of 1024×768 pixels.

The hyperspectral camera, produced by Rikola, is sensitive to the visible and the near infra-red frequency bands and can acquire a full image frame for a given frequency in a single time instant. Multiple frequencies are swept in time. Notice that, due to the UAV motion, the images obtained for different frequencies become misaligned. Up to 25 preprogrammed

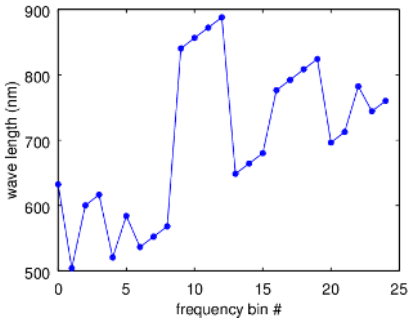


Fig. 6. Sweep order of the frequencies for the hyperspectral camera. The camera acquires sequentially following the bin number but the frequency (and the wavelength) values jump up and down.

frequency bands can be acquired using a frame rate up to one full spectrum image per each two seconds (0.5 Hz). Due to the inner complexity of the camera, the sweep of the frequencies is not monotonic, meaning that the sweep is not performed in an orderly fashion and jumps back and forward between the 25 different frequencies, as shown in Fig. 6.

There is no interface available for the Rikola camera to be controlled by the onboard systems. However, the camera can be pre-programmed to automatically record data into a memory card after an initial time delay. The camera was set and activated before each flight. The same camera parameters were used for all flights. As shown in Fig. 6, wavelengths between 500 nm and 900 nm were acquired.

B. Proposed scenarios and applications

The main usage for the aforementioned system is maritime monitoring. This kind of missions involve scanning large areas of the ocean, that implies long flights which are answered by our aircraft's endurance. Given that in most of the flight's duration there is no significant occurrence, flights become extremely dull for the crew but having an unmanned aircraft allows for the rotation of crews. In this scenario, we consider mainly two broad type of tasks: surveillance/search and environmental monitoring. The first kind of tasks encompasses control of fisheries, detection of smuggling and search of boats in distress. The other kind of activities includes mostly the detection of pollutant substances in the water.

For surveillance missions, we observed different types of objects, spanning from life rafts to cargo ships. The life rafts that were observed had a capacity for 20 people and were 3.7 meters long. The medium size boat that is visible is a patrol boat, 27 meters long. In the sequences, this boat is either stopped or moving at a speed between 4 and 18 knots.

To have an acceptable compromise between the area covered and the perceived detail, the flight's altitude for surveillance/search is in the range of 150 and 300 meters above the ocean surface. Because different lighting conditions demand different technologies, we foresee the use of visible spectrum and LWIR cameras that allow both day and night operation. As mentioned before, these cameras are oriented as shown in Fig. 4, that attempts to capture not only objects near the aircraft with detail but also to detect other at a greater distance.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2775524, IEEE Transactions on Circuits and Systems for Video Technology

To simulate environmental monitoring missions, we flew at an altitude of 100 meters over fish-oil spills dropped to the sea. The fish-oil was selected among several candidates since it presented the closest pattern to the regular boat oil spills and its usage did not pose any particular risk to the environment [11]. In this case, we have used the hyperspectral camera to acquire images.

III. Sequences and Ground Truth Labeling Process

This section describes the method and the developed tool to annotate the video frames with the ground truth bounding boxes and presents information about the video sequences contained in the dataset.

A. Method used

The labeling process consists of marking the position and size of all objects in the images. This is represented as the top left coordinates of a bounding box together with its width and height. Each label corresponds to the smallest rectangular box that contains the object. These rectangles are called bounding boxes and, because they represent the real position of the objects, their set is also called the ground truth. One label is represented by the two coordinates of the top left corner of the bounding box, by its width and height and by an identification number for the object (ID). In this way, multiple objects per image can be accommodated.

The labels have two different purposes. One is to serve as a reference to be used by the training or learning phases of some algorithms, when they learn what an object looks like. These algorithms adjust themselves based on the contents of the bounding boxes. The second purpose is to serve as a comparison reference to evaluate the algorithms. By comparing their object output position with the positions of the bounding boxes an error of some sort can be computed in order to evaluate the algorithm's performance. This means that it is important that these labels are as accurate as possible and this can only be achieved if they are marked manually by a human. The next subsection will describe a tool to facilitate this manual work.

B. The labeling tool

The manual labeling of images is a slow and cumbersome process. There are some labelling tools developed to facilitate this task. One of the most used is [12], a Matlab software package that presents a graphical user interface where users can select and adjust the bounding boxes using a mouse. In periods where the target performs roughly linear trajectories, the user can select just the initial and final bounding boxes the software performs linear interpolation. The user then can adjust each individual bounding box in the interpolated frames using the user interface. Another famous tool is VATIC [13], an online video annotation tool for computer vision research that crowdsources work to Amazon's Mechanical Turk. It has only been tested on Ubuntu with Apache 2.2 HTTP server and a MySQL server. It also presents interpolation functions

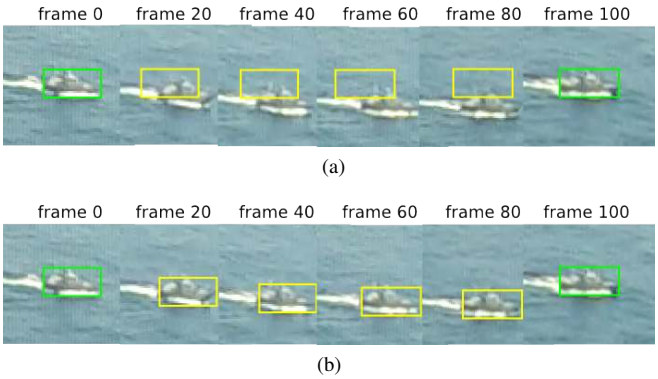


Fig. 7. Examples of the automatic generation of bounding boxes by the labeling tool. (a) Using simple interpolation. (b) Using search after interpolation.

to simplify the the labelling of sections of the video where the movement is linear.

Both [12] and [13] labelling tools present appealing graphical interfaces and general purpose interpolation abilities. However, the requirements for particular software tools and operating systems of these tools make them platform dependent and may prevent their use by some person that wants to work on its own laptop. Furthermore, the interpolation abilities of the existing tools are general purpose and have trouble dealing with our dataset because of the vibrations and constant shaking of the UAV and its cameras.

A new labeling tool was created to overcome these issues. It provides means to speed up the labeler's work and to facilitate it as much as possible, without compromising the labeling quality and ensuring that all of the resulting labels for every frame are in fact manually created, or at least manually verified. Aiming at fast execution and immediate response to user commands, the tool was developed in C++ and only uses the OpenCV library [14] in order to reduce its dependencies and increase its portability[1].

The tool mostly uses keyboard input. The shortcut keys are arranged so that every operation is the most intuitive as possible and requires the least number of key presses as possible (ideally only one).

The tool also provides means to automatically create bounding boxes based on the ones already marked by the user. This includes searching for the object's image in the next frame and interpolating the bounding boxes positions between separated frames (Fig. 7(a)). It can be useful in cases where the object moves linearly across the image. Circular motions can be subdivided into smaller linear sections. The interpolation can also be combined with a local search around the interpolated box position in order to compensate for small camera movements and for movements that are not strictly linear (Fig. 7(b)).

The boxes created automatically are marked as temporary (yellow color boxes in Fig. 7). The user is then forced to go through all of them in order to adjust and mark them as final (green color boxes). If the temporary bounding boxes are already close to the desired positions, then a few keystrokes by the user to adjust them is all that is required. In practice, most

[1]See https://github.com/ricardoarib/labeling_tool for the source code.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2775524, IEEE Transactions on Circuits and Systems for Video Technology

temporary bounding boxes only require small adjustments or do not require any adjustment at all. Nevertheless, the user is still obliged to review them all.

In this way, the tool makes most of the work for the user and still ensures that the labeling is in fact done manually. The tool supports the independent labeling of multiple objects in the images.

C. Sequences statistical data

The dataset presented in this work is composed of 19 video sequences with different properties. In particular, the sequences have different resolutions and durations, the objects are observable for different periods in each sequence and there is a different number of objects in each sequence. Additionally, the sequences were obtained with radiation from different parts of the electromagnetic spectrum. These characteristics are presented in the left part of Table I.

Given the different flight altitudes and camera's perspective, the size of the observed objects varies significantly between different sequences and sometimes in the same sequence. To quantify these variations, Table I also presents the properties of the bounding boxes that encompass the object of interest. These bounding boxes correspond to rectangles, so the average, minimum and maximum size and the standard deviation for the width and height are demonstrated.

The sequences were captured in several scenarios, to improve the robustness of the detectors and trackers. To achieve this, an array of vehicles were used. This set of vehicles was composed by a 27 meters long patrol boat, 90 meters long cargo ships, yachts, Rigid-Hulled Inflatable Boats (RHIBs), life rafts and buoys. In the last column of the mentioned table, the type of objects in each sequence is provided.

IV. Performance Evaluation Methods

Our dataset not only includes the images and the annotations, introduced by a human, that are relevant for the training but we also present the evaluation frameworks that were used for two crucial tasks in surveillance: detection and tracking.

In this section we detail the metrics used for the evaluation of the detection and tracking algorithms, which follow the standard methods currently used in the state-of-the-art.

A. Detection Evaluation Metrics

Traditionally, detection is considered as one of the first operations in a surveillance application. In the case of being automated, this is the cornerstone for all other higher level tasks like tracking or fine-grained classification. In the case of maritime monitoring using airborne images, the detector importance is two fold. On one hand, the detector should be sensitive enough to detect even in a very short observation interval. On the other hand, it should be trustworthy and not overwhelm the operator with false positives (FPs). We will detail how we evaluate these two aspects.

To access the effectiveness of the methods, we have adapted the framework presented by Dollar *et al.*[12]. In the indicated work, a detection is considered as being valid if its bounding box overlaps significantly with a ground truth bounding box. This overlap is measured with the intersection over union (IoU) calculated as

$$\text{IoU} = \frac{\text{area}\{\bar{D}^t \cap G^t\}}{\text{area}\{\bar{D}^t \cup G^t\}}. \tag{1}$$

Dollar *et al.* require IoU to be larger than 50% to consider a detection as valid. In our experiments, because the bounding boxes for ground truth and detections are small when compared with the image, any small error in localization or size, has a big impact on IoU. Requiring the IoU to be larger than 50% in small object detections will result in many objects being reported as false positives when they actually are true positives, thus providing biased evaluations of the detection algorithms. To make the evaluation fairer for the conditions of our dataset, we have tested several threshold values to match a detection to a ground truth, in particular we have evaluated using IoU $> 0\%$, IoU $> 10\%$ and IoU $> 20\%$.

Having defined the matching method, we use two main metrics to quantify the performance: Precision and Recall. The first measures if the detections being produced are relevant to the problem. Detections can be aggregated into correct (true positive (TP)) and incorrect detections (false positive (FP)). This metric is computed as the ratio of the correct detections over the complete set, *i.e.* $Precision =$ # TP/ (# TP + # FP) .

Recall gauges the portion of TP over the entire set of ground truth labels (TPs and false negatives (FNs)) and is calculated as $Recall =$ # TP/ (# TP + # FN) .

The mentioned metrics characterize a given operating point of an algorithm but many detectors provide a score for its output and therefore the user can select the threshold to consider a detection. This can create an infinite number of operating points. To overcome this issue, we plot Precision-Recall (PR) values by ranging the detection score's threshold from the minimum to the maximum. Operating points that correspond to the minimum score typically result in high Recall. Conversely, points corresponding to high scores usually result in high Precision.

Despite the qualities of a PR curve $p(r)$, when comparing several detectors it is useful to have a quantity that encapsulates the overall performance. One common metric is to use the Mean Average Precision (mAP) that is defined as

$$mAP = \int_0^1 p(r)dr \tag{2}$$

and may be approximated by the Area Under the Curve (AUC) which is computed as the sum of Precision $p(k)$, at every possible threshold with the index k, times Recall's variation $\Delta r(k)$ between these points.

$$AUC = \sum_{k=1}^{N} p(k)\Delta r(k) \tag{3}$$

B. Tracking Evaluation Metrics

For the purpose of tracking, it is important to evaluate how well the system can keep track of the object without losing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2775524, IEEE Transactions on Circuits and Systems for Video Technology

TABLE I
VIDEO SEQUENCE AND OBJECTS CHARACTERISTICS

						objects width			objects height			
Name	Spectrum	Resolution	Frames	Labels	Objects	Ave.	Range	Std. Dev.	Ave.	Range	Std. Dev.	Type of boat/object
seq01	Vis.	$640 * 480$	102707	891	4	15	11-91	6	26	11-91	17	Patrol, RHIB, life raft
seq02	Vis.	$1024 * 768$	16369	19621	2	21	4-84	18	18	2-63	14	Patrol, Life Raft
seq03	LWIR	$384 * 288$	7090	1764	1	38	9-56	12	24	3-46	12	
seq04	Vis.	$1920 * 1080$	300	519	2	34	56-69	28	20	6-34	8	Patrol; buoy
seq05	Vis.	$1920 * 1080$	1080	2160	2	32	6-78	26	26	6-34	17	Patrol; buoy
seq06	Vis.	$1920 * 1080$	4860	8426	2	36	6-128	31	35	2-100	24	Patrol; buoy
seq07	Vis.	$1920 * 1080$	720	739	3	13	5-16	1	24	5-33	6	Sailing yacht
seq08	Vis.	$1920 * 1080$	1440	236	3	15	6-5	5	12	5-17	3	Yacht
seq09	Vis.	$1920 * 1080$	480	358	1	45	14-52	5	19	13-23	2	Yacht; Patrol
seq10	Vis.	$1920 * 1080$	420	378	1	29	3-36	5	15	9-20	2	Patrol
seq11	Vis.	$1920 * 1080$	5880	5185	1	46	12-94	17	44	4-75	15	Patrol
seq12	Vis.	$1920 * 1080$	1276	1276	1	26		2	11		3	Cargo
seq13	Vis.	$1920 * 1080$	2251	2044	2	18	9-41	4	20	6-33	5	Cargo
seq14	Vis.	$1920 * 1080$	506	1008	2	27	8-60	17	16	7-34	9	Patrol
seq15	Vis.	$1920 * 1080$	1071	941	1	97	21-156	34	45	21-70	7	Patrol
seq16	Vis.	$1920 * 1080$	1401	1237	1	111	4-216	52	51	8-84	13	Patrol
seq17	Vis.	$1920 * 1080$	751	504	1	47	18-81	15	40	16-63	9	RHIB
seq18	Vis.	$1920 * 1080$	2251	1121	1	45	19-69	11	38	12-59	9	RHIB
seq19	Hyperspec.	$1024 * 648$	900	(unavailable)[2]	2	-	-	-	-	-	-	Patrol Pollutant

it. The tracker is given the initial position of the object, either manually or by some detection system, and autonomously tries to follow the movements of the target object.

Different metrics are required for the tracking case and we propose the use of the Object Tracking Benchmark (OTB) [15] framework for this dataset. This methodology evaluates the tracking methods by computing Precision and Success plots of the tracking under two different initialization strategies denoted Temporal Robustness Evaluation (TRE) and Spatial Robustness Evaluation (SRE).

The Precision plot shows the percentage of frames whose Euclidean distance between the centers of the detection and the manually labeled ground truths is lower than a given threshold. The Precision threshold varies from 0 to 50 pixels with a step of 1 pixel. The score chosen to rank the trackers is Precision value for a threshold of 20 pixels as suggested by [15].

The Success plot evaluates the bounding box overlap of the detection with the ground truth as defined by (1). It shows the percentage of frames whose bounding box overlap ratio IoU is higher than a given threshold from ratio values of 0 to 1, where 1 means perfect match of the detection and ground truth and 0 meaning lost target. To rank different algorithms, the AUC of each Success plot is used.

The Temporal Robustness Evaluation consists in initializing the trackers at different frames, not just the first, and running them until the end of the sequence. Each sequence is evaluated by initializing in 20 different frames. The initial frames are chosen by starting with the first frame of the sequence and stepping through them at a regular interval. The step is approximately the number of frames of the sequence divided by 20.

The Spatial Robustness Evaluation consists in introducing error in the initialization by shifting the bounding box by 10% of the target size in 8 different directions and scaling it by 0.8, 0.9, 1.1 and 1.2 of the ground truth size. This results in 12 different initialization.

These evaluations are pertinent because in a real world scenario the trackers would be initialized with a vessel detector that is likely to introduce error in the initialization.

V. BENCHMARK RESULTS FOR THE DATABASE

In this section, selected detection and tracking methods are applied to the dataset sequences. These methods were developed by the authors in prior research [11], [16], [17], [18] and their results are compared with other state-of-the-art approaches. The results define a baseline performance of detection and tracking methods to be used as reference for future research.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2775524, IEEE Transactions on Circuits and Systems for Video Technology

TABLE II
EXPERIMENTS PERFORMED FOR BENCHMARKING

Experiment type	Camera spectrum	Sequences used
Boat detection	Visible	*seq01*, *seq02*, *seq06*, *seq12*, *seq13*, *seq14*, *seq15*, *seq16*
Boat tracking	Visible	*seq01*, *seq02*
Pollutants detection	Hyperspectral	*seq19*

Three different problems are considered: the detection of boats, where the purpose is to identify if there is a boat in the image and where it is located; the tracking of the boat, where the movement of the boat follows an initial detection; and lastly the detection of pollutants using hyperspectral images. A summary of all experiments identifying the video sequences used as well as the types of cameras used is shown in Table II.

A. Detection results

Four detection algorithms have been applied to the visible spectrum sequences, to create baseline results. The methods that were used, differ in its nature with two unsupervised and two supervised methods. The supervised methods need to be trained and therefore some sequences have been chosen as training set and others as testing set. We have used all the frames present in sequences *seq01*, *seq12*, *seq14* and *seq15* for the training stage, with 10% of those images used as validation set. For the testing stage, we have considered different sequences, to avoid having very similar images in the train and test set, hence we have used all the frames in sequences *seq02*, *seq06*, *seq13* and *seq16*. Among the test sequences, we have used *seq02* and *seq06* to carry out a qualitative evaluation and for a quantitative evaluation, we have used *seq13* and *seq16*.

The first unsupervised method that was tested was *Image Signature Saliency Method* [19]. Using this standard saliency method without any modifications, resulted in very poor results and, therefore, its results are not reported. This fact supports the existence of challenging conditions in this dataset and indicates that specialized detection methods are needed. The second detection technique that we have tested is denominated as *Blob Algorithm* [20]. This method is composed of three stages: Vessel Detection, Spatial Detection and Time Consistency. Vessel Detection first looks for salient pixels, *i.e.* with high intensity and color features, creating a binary map where *zero* represents background and *one* represents areas of interest. Spatial Detection creates regions labeled as *boat* or *background*, by applying morphological operations to the binary image and rules regarding the size and position of the regions, to eliminate large regions or regions adjacent to the image border (typically caused by sun glare and sky, respectively). Both the morphological operations and the rules must be tuned for specific scales of the boat and the glare. The last stage (Time Consistency) creates a buffer of several time instants and only blobs that are persistent through time are considered as valid.

[2]Due to the difficulty to visually identify the oil spill, there is no ground truth available for the hyperspectral sequence.

The third method, dubbed *Blob+CNN* [16], is partially based on *Blob Algorithm* and partially on Convolutional Neural Networks (CNN). This method uses the first stage of the *Blob Algorithm*, to create a binary map identifying possible regions of interest. Afterwards, these regions are cropped and used as object proposals. These proposals are then provided to a standard CNN to be classified as *boat* or *background*. This method is similar to R-CNN [21] as it creates patches of images that are fed into a neural network. However, *Blob+CNN* was designed to be deployable on embedded platforms, which has enforced the creation of a small number of proposals. The network that is used is alexnet and was retrained with images from the already mentioned training sequences.

The fourth method was *detectnet with Multiple Hypothesis Tracker (MHT)* [17]. This method uses a convolutional network (inspired by the ideas presented in [22]) to generate detection bounding boxes and then used a Multiple Hypothesis Tracker to associate detections in successive time instants. The detection network creates a grid that indicates if a given region of the image is a *boat* or *background* and also computes a regression for the bounding box containing the boat. If a boat is present in multiple cells of the grid, the bounding boxes are merged. In this case, the goal of the MHT is to verify time consistency by computing the probability of a given set of consecutive detections being generated by a real boat.

In Fig. 8, we have two sequences and three IoU conditions. Note that the IoU does not have influence on the quality of the detections bounding boxes but rather on the reported performance. With the least demanding condition (IoU > 0), any overlap larger than 0 between ground truth and detection bounding box is enough to consider a detection as correct. In this case, almost every detection is considered as positive detection which causes the Precision to be close to 1. Nonetheless, there are some time instants where no detection is produced and precludes Recall from attaining 1. With these ideas in mind, we verify that the *Blob Algorithm* fails to produce the same amount of bounding boxes (even with relaxed location and size restrictions) as the other two methods. In the two other evaluation conditions (IoU > 10 and IoU > 20), the location and the size of bounding boxes must be more adjusted for a detection to be considered as correct. With these conditions, the *Blob+CNN* method achieves a high Precision at the cost of only accepting detections with a very high score[3] and discarding many lower score detections. Consequently, a low Recall is obtained by this method. *Detectnet with MHT* maintains a higher Precision without sacrificing as much Recall. This is the algorithm that gets closer to the ideal condition (Precision $= 1$ and Recall $= 1$) and this holds true for different evaluation criteria, *i.e.* for IoU > 0, IoU > 10 and IoU > 20 and also for both sequences (*seq13* and *seq16*).

The poorer performance of *Blob Algorithm* is caused by the lack of discriminative power of the vessel detector stage to distinguish between boats and distractors. This compels this stage and the following to discard many occurrences (some of which are correct), otherwise the number of false detections

[3]This score is the "confidence" that a detector has in each detection, as mentioned in subsection IV-A.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2775524, IEEE Transactions on Circuits and Systems for Video Technology

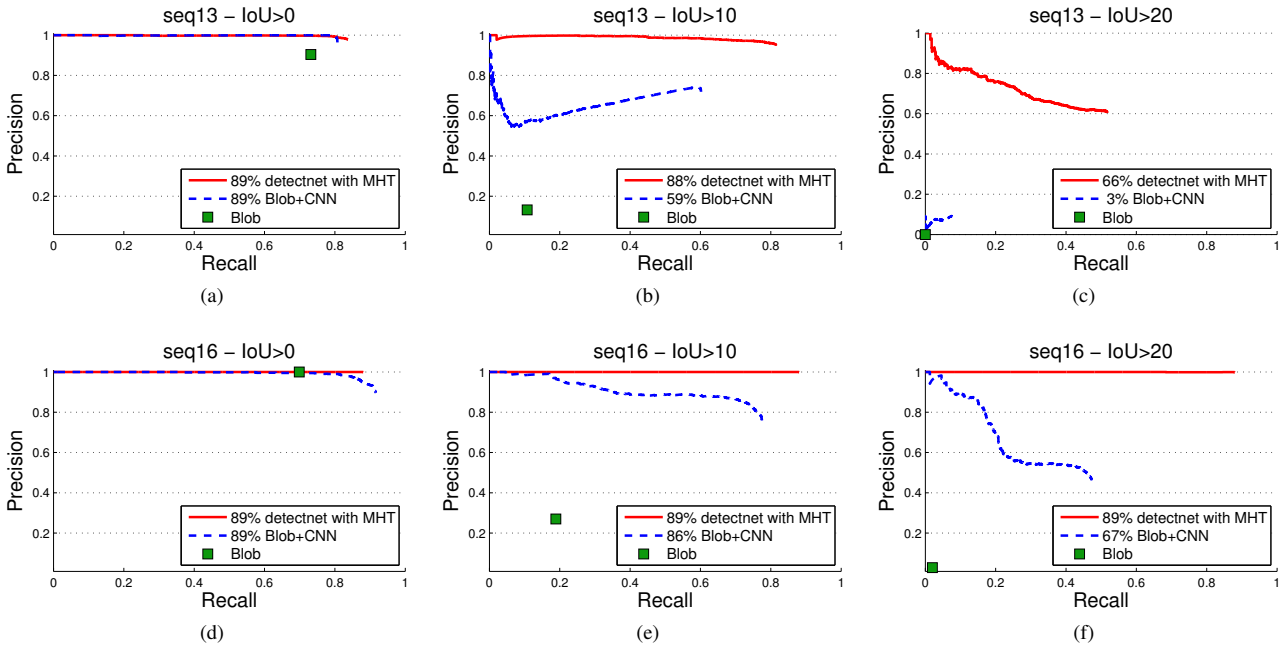


Fig. 8. Precision-Recall (PR) curves obtained with *seq13* (top row) and *seq16* (bottom row). The results for each sequence were evaluated with three conditions: IoU $> 0\%$, IoU $> 10\%$ and IoU $> 20\%$. Each sequence was processed with three different algorithms (*detectnet with MHT*, *Blob+CNN* and *Blob Analysis*). The first two methods have a variable operating point, which resulted in PR curves and have the AUC presented in the legend. The Blob method only has one operating point, which results in only one Precision-Recall condition.

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

Fig. 9. Example of bounding boxes obtained with detectnet+MHT. In (a) a life raft and in (b) a patrol boat with a lift raft are detected, despite the small size and the presence of glare and sky on the images. In (c) the patrol boat is detected inside an area with glare. Here, the bounding box is not completely adjusted to the boat and is one of the cases that is considered as false positive when the evaluation criteria is more demanding (IoU $> 10\%$ and $> 20\%$). In (d) the boat is detected with an appropriate bounding box and no false positives are present on the image.

would be overwhelming. Even though *Blob Algorithm* and *Blob+CNN* share the same approach to create the initial blobs, there is a significant difference in the capability of the CNN to separate distractors from boats, which explains the difference between the two methods. Nonetheless the first stage is tuned for a given scale and therefore the size of the bounding boxes is not perfectly adjusted to the varying size of the objects. The mismatch in size creates more false positives as the overlapping criterion gets more demanding and is especially severe in *seq13* that has a smaller scale. Unlike typical scoring functions, which assign high values to true positives and low values to false positives, on some occasions, *Blob+CNN* incorrectly assigns high values to TP and low values to FP. This causes the initial decrease followed by an increment of the *Blob+CNN* plot in Fig. 8(b) and Fig. 8(c). In *detectnet with MHT*, only the objects' minimum size is specified and the merging of bounding boxes allow objects with bigger size to be detected. Furthermore, the coherence between consecutive detections is verified using the MHT, which proved to discard most spurious events like momentary glare or wave crests.

Additionally, to further test the best approach, we have also tried this method in sequences *seq06* and *seq02*. In these sequences, the detection of buoys and life rafts, as presented in Fig. 9(a) and Fig. 9(b), is only occasionally successful. Another very challenging condition is when the boats appear inside regions with glare, as presented in Fig. 9(c). Some detection results are shown in Fig. 9. These images correspond to very challenging conditions. In most of these images, the scale of the object of interest is small when compared to the image and there is strong glare.

B. Tracking results

Videos from this database were used to evaluate different state-of-the-art tracking methods using the framework pro-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2775524, IEEE Transactions on Circuits and Systems for Video Technology

posed above. The tested trackers were ASMS [25], DSST [26], KCF [27], CF2 [24], SRDCF [28], MUSTer [29], MEEM [30], MDNet [23]. These methods are generic and do not take into account the specific difficulties that these maritime scenario images impose and new methods [18] were developed, identified as OURS and OURS_CNN and which we will describe next, aiming at this kind of scenarios. The new methods are based on kernelized correlation filters.

Correlations filters are templates specially tuned to a particular image pattern. The correlation of this template with the new image x produces a response map y:

$$y = x * h^{-}, \tag{4}$$

where h^{-} is the reflection in both coordinates of the patch and $*$ is the convolution operator. The location corresponding to the maximum value of the response map will indicate the new position of the target. To improve the computational efficiency of the filter, the convolution is computed in the Fourier domain:

$$y = \mathcal{F}^{-1}(\hat{x} \odot \hat{h}^{*}), \tag{5}$$

where the hat symbols represent the discrete Fourier transform of the vectors, $\mathcal{F}^{-1}$ represents the inverse Discrete Fourier Transform, $\odot$ represents the element-wise multiplication, and the superscript $*$ indicates the complex conjugate.

The tracker uses a patch x of the image containing the object to learn the filter coefficients h so that the filter produces a particular response y having a peak at the center of the patch. Typically, y is defined as an isotropic Gaussian function with a small standard deviation. A simple way to compute an exact filter is proposed in [31] using the Fourier domain:

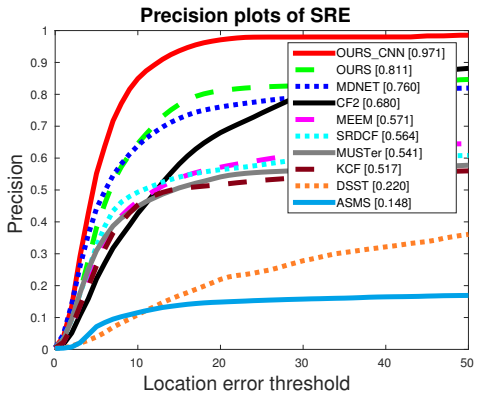
$$\hat{h}^{*} = \frac{\hat{y}}{\hat{x}} \tag{6}$$

The filter performance can be improved by using a kernelized version of it that maps the image into a higher order space. Taking advantage of the properties of circulant matrices [27], the filter computation can be expressed in the following computationally efficient form

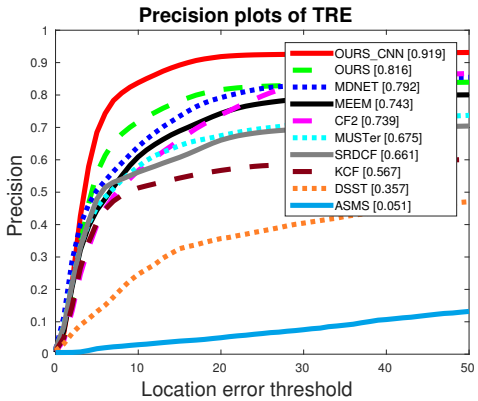
$$\hat{h} = \frac{\hat{x}^{*} \odot \hat{y}}{\hat{x}^{*} \odot \hat{x} + \lambda}, \tag{7}$$

where the division, as well as all other operations, is element-wise.

The new methods we propose update the correlation filter for each new image in order to comply with the variations of

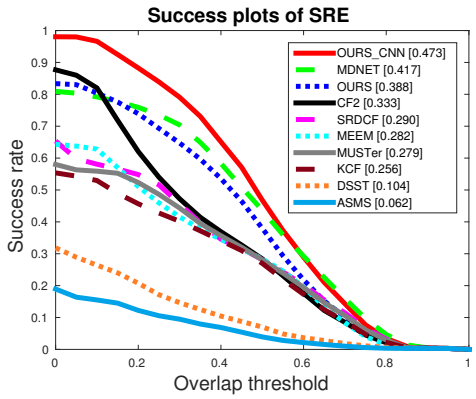


(a)

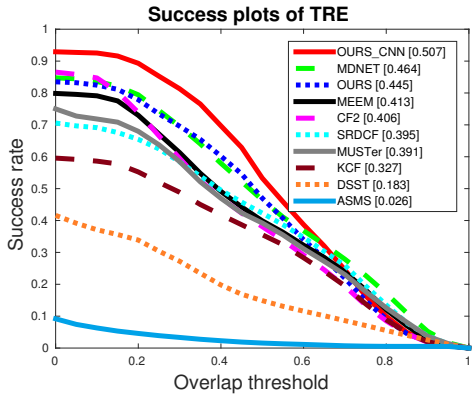


(b)

Fig. 10. Precision plots for the (a) SRE and (b) TRE from the OTB framework [15]. The values on the legend correspond to the Precision for a location error threshold of 20. Note that the colors represent the ranking and not the different trackers. The same tracker can have different colors in different evaluations depending on its ranking (e.g. red corresponds to the first place, green to the second place and blue to the third place).



(a)



(b)

Fig. 11. Success plots for the (a) SRE and (b) TRE from the OTB framework [15]. The values on the success plots legend correspond to the areas under the curves (AUC). Note that the colors represent the ranking and not the different trackers. The same tracker can have different colors in different evaluations depending on its ranking (e.g. red corresponds to the first place, green to the second place and blue to the third place).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2775524, IEEE Transactions on Circuits and Systems for Video Technology

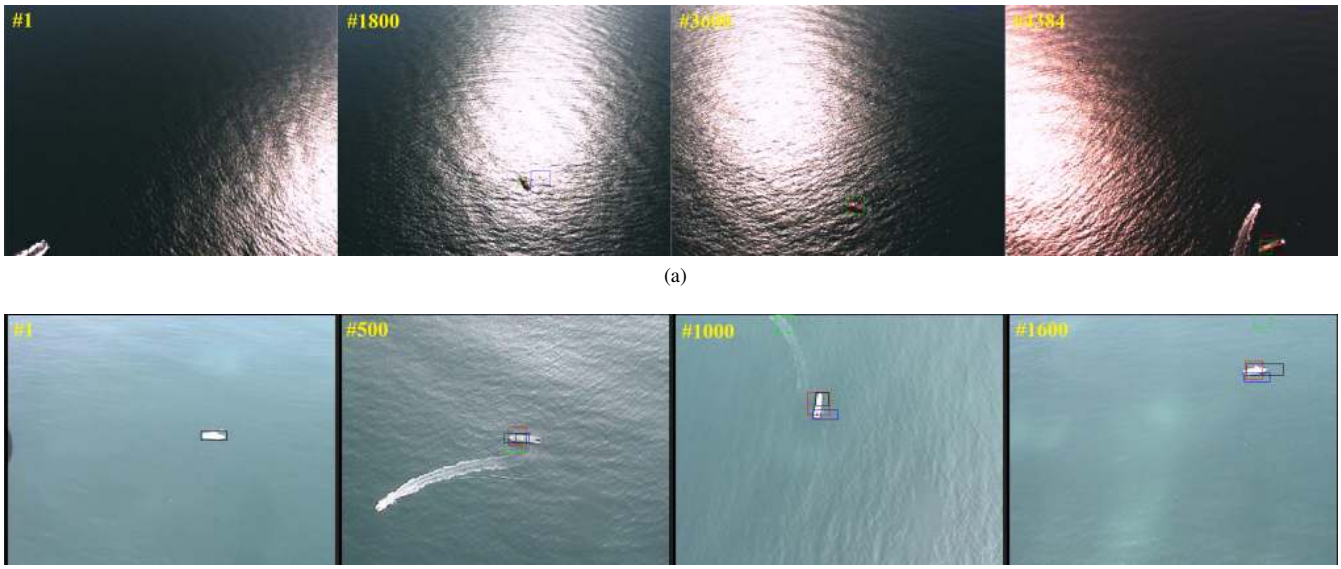


Fig. 12. Frames from the data set with the results, as bounding boxes, of some of the best performing methods: both our methods [18] (OURS as green and OURS_CNN as red), MDNet [23] (black) and CF2 [24] (blue). Two examples are shown. Example (a) uses *seq02* and example (b) uses *seq01*.

the boat's appearance over time, thus making the algorithms robust to rotations of the boat relative to the camera point of view or to illumination variations like when the boat encounters sun reflections. However, this introduces drift problems and the algorithms tend to lose the target after some time. To avoid drift effects, an additional adjustment based on blob analysis was introduced, being only active if the conditions are right.

Before the detection of the connected components, usually referred to as blobs, the image patch of interest has to be segmented. We adopt a binarization approach via the Otsu's method [32] to separate bright and dark parts of the image that, in principle, will correspond to the boats and the ocean surface, respectively. An erosion step then isolates the vessel from nearby distractors like wakes, or other vessels, and also removes some noise originated from waves and sun reflections.

The segmented pixels are then grouped in blobs that correspond to the maritime vessels. However, the background clutter (waves and wakes) and the sun reflections can interfere with the segmentation. To circumvent this problem, a set of heuristics were defined to determine if the conditions allow for the track correction. If a) no blob touches the border of the region of interest (ROI); b) the number of blobs is smaller or equal to a defined threshold T_n and c) the blob has an area larger than a second threshold T_s, then the conditions are favorable to correct the track. The first and second conditions are used to detect the presence of sun reflections or background clutter as boat wakes or waves, and the third is used to filter some noise and boat wakes that might go through the two first filters. If the conditions are met, then a blob is chosen using a nearest-neighbor approach with regard to the latest tracking position which is then updated to the same position as the center of that blob.

Also, instead of using the images x directly, the methods use features extracted from those images. These can either be the HoG features, as used in method OURS, or be CNN features, as used in method OURS_CNN.

The new methods can achieve better tracking performance as is shown in Fig. 10 and Fig. 11 using the metrics defined in the previous section for tracking. Also, Fig. 12 shows some examples of detections marked on the images. The example in Fig. 12(a) is a case of success for both our methods given that the track is still on the target after 4383 frames (when the target leaves the camera's field of view), even after going through a region with intense sun-reflections multiple times. The other top performing algorithms lose the target much earlier. Fig. 12(b) shows a case of failure for our method using HoG features, where the track gets lost following the wake of another target. Nevertheless, our method using CNN features can overcome this problem given its capability to better discriminate the target from the wakes.

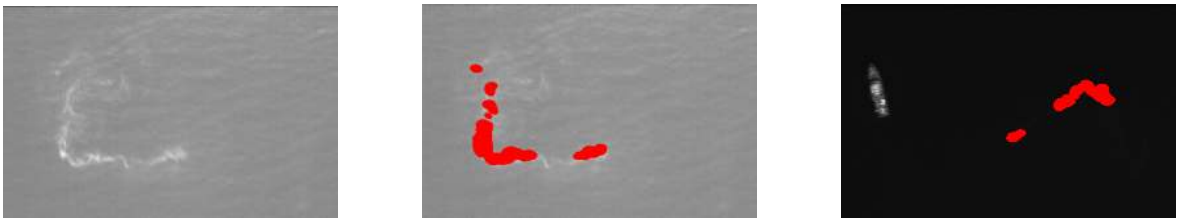
Full details about the new tracking algorithms and a detailed analysis of the results shown here can be found in [18].

C. Hyperspectral results

The analysis of hyperspectral images captured revealed that the spectrum of the image is different for the oil spill when compared to the spectrum of the unpolluted water. This suggests that spectral based methods should be successful to identify the oil spill. However, since there is misalignment of the images for different spectral bands, state of the art methods to analyse the hyperspectral content, such as spectral signature matching [33] or end-member extraction methods [34] cannot be applied in a straightforward manner.

We proposed a new method [11] based on simple logic rules and using morphological transforms based on the erosion and dilation operators [35]. Three frequency bands, not very far apart in acquisition time, were identified to contain information about the presence of an oil spill. The spectral signature is simply the level of two of them being higher than the third.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2775524, IEEE Transactions on Circuits and Systems for Video Technology



(a) (b) (c)

Fig. 13. Oil spill detections using hyperspectral images. (a) Shows the oil spill. (b) Shows the detection. (c) Shows the detection when in the presence of a boat. The images in (a) and (b) have their luminance increased in order for the oil spill to be visible while the image in (c) is shown in its original luminance.

From the normalized level comparisons, we get two blobs that are slightly eroded to clear noise. Afterward, they are dilated by an amount large enough so that the two blobs can still overlap despite the misalignment of the images. The intersection of booth blobs is considered to be an area of positive detection of the oil spill. Boats and other objects, which have a much stronger luminance when compared to the oil, are eliminated from the detections using an additional threshold rule.

Fig. 13(a) shows an image for one frequency channel where the oil spill is present. The image luminance was increased in order for the spill be clearly visible. Fig. 13(b) shows the detection of the spill area by the new method. Fig. 13(c) shows the detection for another image where a boat is also present. In this case, the normal luminance is shown. We can see that the oil spill is detected and not the boat, showing the robustness of the method to the presence of other objects in the water.

More details about the oil spill detection method can be found in [11].

VI. Conclusion

In the present work, we have described a dataset of properly labeled images, captured by a small aircraft in maritime surveillance scenarios. To the best of our knowledge, this is the first publicly available dataset in such scenario. Given that many of the current computer vision and pattern recognition methods (such as deep learning) are data-driven, we believe that this is a strong contribution to allow the training of new methods.

To provide more context to the imagery in the dataset, the conditions and the system were described. Also, we described the content information of the labels and introduced a new labeling tool that allowed us to annotate so many images.

Using standard evaluation frameworks over this dataset data, we also present some baseline results for state of the art detection and tracking methods, and for some methods developed for maritime scenarios by the authors. In this way, a starting point is established as a baseline for the comparison with any future methods.

The results for an hyperspectral method, which to our knowledge is unique for these scenarios, are also included.

For future work, we will proceed in two directions. In one direction we will address the use of the dataset for applications beyond detection and tracking, for example spatiotemporal event detection (boats approaching/departing, boats navigating side by side, boats driving fast), and exploit recent algorithms on one/zero shot learning of object categories/events. On the other direction, we will enrich the dataset with more labels to describe the context of the frame (with sun glare, with wave crests, etc) which will allow discriminating the performance of methods across different contexts. This will allow a finer analysis of the advantages and disadvantages of algorithms in the specific challenges of maritime scenarios.

Acknowledgment

This work was partially supported by ANI project SEAGULL (QREN SI IDT 34063), FCT project SPARSIS [PTDC/EEIPRO/0426/2014], and FCT project [UID/EEA/50009/2013]. The authors would like to thank all the people from VisLab and Portuguese Air Force Research Center that allowed the collection and labeling of the video sequences.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[2] D. Figueira, M. Taiana, A. Nambiar, J. Nascimento, and A. Bernardino, "The hda+ data set for research on fully automated re-identification systems," in *European Conference on Computer Vision*. Springer, 2014, pp. 241–255.

[3] A. Nambiar, M. Taiana, D. Figueira, J. Nascimento, and A. Bernardino, *A Multi-camera video data set for research on High-Definition surveillance*, 2014.

[4] D. D. Bloisi, L. Iocchi, A. Pennisi, and L. Tombolini, "ARGOS-Venice boat classification," in *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*, 2015, pp. 1–6.

[5] M. M. Zhang, J. Choi, K. Daniilidis, M. T. Wolf, and C. Kanan, "Vais: A dataset for recognizing maritime imagery in the visible and infrared spectrums," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 10–16.

[6] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 187–203, 2016.

[7] K. Liu and G. Mattyus, "Fast multiclass vehicle detection on aerial images," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 9, pp. 1938–1942, 2015.

[8] L. Patino, T. Cane, A. Vallee, and J. Ferryman, "Pets 2016: Dataset and challenge," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 1–8.

[9] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–24, 2017.

[10] R. Ribeiro, "Seagull Database Web Page," http://vislab.isr.tecnico.ulisboa.pt/seagull-dataset/, 2016, [Online; accessed 01-January-2017].

[11] M. M. Marques, V. Lobo, R. Batista, J. Almeida, M. d. F. Nunes, R. Ribeiro, and A. Bernardino, "Oil spills detection: Challenges addressed in the scope of the seagull project," in *OCEANS 2016 MTS/IEEE Monterey*, Sept 2016, pp. 1–6.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2775524, IEEE Transactions on Circuits and Systems for Video Technology

[12] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 4, pp. 743–761, 2012.

[13] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently scaling up crowdsourced video annotation - a set of best practices for high quality, economical video labeling," *International Journal of Computer Vision*, vol. 101, no. 1, pp. 184–204, 2013.

[14] "OpenCV Library," http://opencv.org, [Online; accessed 04-October-2017].

[15] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[16] G. Cruz and A. Bernardio, "Aerial detection in maritime scenarios using convolutional neural networks," in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2016, pp. 373–384.

[17] G. Cruz and A. Bernardino, "Evaluating aerial vessel detector in multiple maritime surveillance scenarios," in *OCEANS 2017 MTS/IEEE Anchorage*, 2017, accepted.

[18] J. Matos, A. Bernardino, and R. Ribeiro, "Robust tracking of vessels in oceanographic airborne images," in *OCEANS 2016 MTS/IEEE Monterey*, Sept 2016, pp. 1–10.

[19] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 1, pp. 194–201, 2012.

[20] J. S. Marques, A. Bernardino, G. Cruz, and M. Bento, "An algorithm for the detection of vessels in aerial images," in *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*. IEEE, 2014, pp. 295–300.

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[23] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," *arXiv preprint arXiv:1510.07945*, 2015.

[24] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3074–3082.

[25] T. Vojir, J. Noskova, and J. Matas, "Robust scale-adaptive mean-shift for tracking," *Pattern Recognition Letters*, vol. 49, pp. 250–258, 2014.

[26] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.

[27] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *Pattern Analysis and Machine Intelligence, IEEE Trans.*, vol. 37, no. 3, pp. 583–596, 2015.

[28] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4310–4318.

[29] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 749–758.

[30] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: robust tracking via multiple experts using entropy minimization," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2014.

[31] D. S. Bolme, , B. A. Draper, and J. R. Beveridge, "Average of synthetic exact filters," in *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*. IEEE, 2009, pp. 2105–2112.

[32] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[33] M. S. Alam and P. Sidike, "Trends in oil spill detection via hyperspectral imaging," in *2012 7th International Conference on Electrical and Computer Engineering*, Dec 2012, pp. 858–862.

[34] D. Sykas, V. Karathanassi, C. Andreou, and P. Kolokoussis, *Oil spill mapping using hyperspectral methods and techniques.*, 2012.

[35] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

Ricardo Ribeiro (Ph.D. 2012) received the Ph.D. degree in electrical and computer engineering from the Instituto Superior Técnico (IST), faculty of the Lisbon University. Currently is a research fellow of the Computer and Robot Vision Lab (VisLab) in the Institute for Systems and Robotics at IST. He has participated in several national and international projects, both as a researcher and as a technical advisor, in the areas of video surveillance, drones, and assistive robotics. He is also supervisor and co-supervisor of M.Sc. and Ph.D. theses. His research interests include the application of signal processing, control theory, computer learning, computer vision and sound processing to advanced robotics and human-computer interaction systems.

Gonçalo Cruz (M.Sc. 2012) received the M.Sc. degree in electrical and computer engineering from the Portuguese Air Force Academy, Sintra, Portugal and he is also a Ph.D. student at the Instituto Superior Técnico, Lisbon University, Portugal. He is a lecturer with the Portuguese Air Force Academy and a researcher with the Portuguese Air Force Research Center, working in the area of computer vision applied to unmanned aerial vehicles. His research interests include machine learning and computer vision applied to aerial robotics.

Jorge Matos (M.Sc. 2016) studied Electrical and Computer Engineering at IST, Lisbon University. He has a major in Control and Decision Systems and minor in Computer Sciences. His master thesis had a focus on computer vision and machine learning with the goal of developing tracking algorithms in the context of oceanographic airborne imagery. Currently working in the industry of ITS (Intelligent Transportation Systems), his main interests are machine learning, deep learning, and computer vision.

Alexandre Bernardino (Ph.D. 2004) is an Associate Professor at the Dept. of Electrical and Computer Engineering and Senior Researcher at the Computer and Robot Vision Laboratory of the Institute for Systems and Robotics at IST, the faculty of engineering of Lisbon University. He has participated in several national and international research projects as principal investigator and technical manager. He published more than 40 research papers in peer-reviewed journals and more than 100 papers on peer-reviewed conferences in the field of robotics, vision and cognitive systems. He is associate editor of the journal Frontiers in Robotics and AI and of major robotics conferences (ICRA, IROS). He has graduated 10 Ph.D. students and more than 40 M.Sc. students. He was co-supervisor of the Ph.D. Thesis that won the IBM Prize 2014 and the supervisor of the Best Robotics Portuguese MSc thesis award of 2012. He is the current chair or the IEEE Portugal Robotics and Automation Chapter. His main research interests focus on the application of computer vision, machine learning, cognitive science, and control theory to advanced robotics and automation systems.