Multidim Syst Sign Process
https://doi.org/10.1007/s11045-018-0558-4

Anomaly detection with a moving camera using multiscale video analysis

Gustavo H. F. de Carvalho[1] · Lucas A. Thomaz[1] · Allan F. da Silva[1] · Eduardo A. B. da Silva[1] · Sergio L. Netto[1]

Received: 18 December 2017 / Revised: 20 February 2018 / Accepted: 22 February 2018

Abstract This paper addresses the problem of abandoned object detection in a cluttered environment using a camera moving along a straight track. The developed system compares captured images to a previously recorded reference video, thus requiring proper temporal alignment and geometric registration between the two signals. A real-time constraint is imposed onto the system to allow an effective surveillance capability in practical situations. In this paper, we propose to deal with the simultaneous detection of objects of different sizes using a multiresolution approach together with normalized cross-correlation and a voting step. In order to develop and properly assess the proposed method we designed a database recorded in a real surveillance scenario, consisting of an industrial plant containing a large number of pipes and rotating machines. Also, we have devised a systematic parameter tuning routine that allows the system to be adapted to different scenarios. We have validated it using the designed database. The obtained results are quite effective, achieving real-time, robust abandoned object detection in an industrial plant scenario.

Keywords Video surveillance · Moving camera · Abandoned object detection · Cluttered environment

1 Introduction

Computer vision techniques can be used to extract reliable information from a video signal, providing a reasonably good understanding of the recorded environments or processes. By exploring this capability, automatic video-based monitoring systems can be deployed, enabling significant savings of resources, minimizing labor risks in hazardous environments and increasing system efficiency, particularly when dealing with repetitive or tedious tasks.

✉ Lucas A. Thomaz
lucas.thomaz@smt.ufrj.br

[1] Electrical Engineering Program, Federal University of Rio de Janeiro, PO Box 68504, Rio de Janeiro, RJ 21941-970, Brazil

Published online: 28 February 2018

The automatic detection of abandoned objects in a given scenario constitutes an interesting feature of a surveillance or remote inspection system. This detection problem can be addressed by comparing a newly acquired video, also known as the target video, to a reference video considered free of abandoned objects. In this way, a video anomaly, which may be associated to an abandoned object, is detected whenever and wherever the target and reference videos differ to a significant amount.

When using static cameras for this purpose, simple background and behavior subtractions, employing statistical approaches, allow one to detect anomalies in the acquired videos (Dore et al. 2010; Subudhi et al. 2011; Saligrama et al. 2010; Tian et al. 2011). A good example of such systems is described in Lin et al. (2017), where the authors deal with a background with frequent local motions, such as waving trees. Also, in Chang et al. (2013), the proposed method tries to identify the owner whenever it detects an abandoned object.

Some works relax the static camera constraint by detecting moving objects while compensating a small amount of movement, usually due to some kind of jitter in the camera (Cheng et al. 2011; Jodoin et al. 2012; Romanoni et al. 2014). Another popular application of automatic surveillance systems is to track objects along a video sequence. As in the case of object detection applications, this task is performed more easily when static cameras are employed (Kim and Hwang 2002; Subudhi et al. 2011). Some other methods rely on a pre-trained database of object shapes to detect the presence of anomalous objects in the scene (Felzenszwalb et al. 2010).

A surveillance system based solely on static cameras, however, may not be efficient in cases where a wide area must be supervised (Tomioka et al. 2012) or expensive specialized cameras [e.g. infrared, hydrocarbon-detecting cameras, like FLIR GF320 model (FLIR 2016)] are employed.

A possible solution in some of these situations is to use pan-tilt-zoom (PTZ) or panoramic cameras that add some flexibility to the camera field-of-view (FoV). Some approaches detect mobile objects with PTZ cameras by building a dynamic background model and applying modified background subtraction techniques (Micheloni and Foresti 2006; Suhr et al. 2011; Xie et al. 2010; Xue et al. 2013, 2011). Others, such as Davis et al. (2007), develop motion histograms and detect events whose motion differs from that estimated for the camera.

Another solution, particularly suited for specialized cameras, is the use of a moving platform to increase the surveillance range. Such solution, however, brings new challenges as the camera movement must be properly compensated in time and space before any sort of comparison between the target and reference videos can be made (Hu et al. 2015; Kong et al. 2010).

Many works have addressed the problem of detecting a moving object using a moving camera. Some Kim et al. (2010), Nordlund and Uhlin (1996), Pinto et al. (2014) and Sun et al. (2013) approach this challenge by using optical flow techniques and building several background models to deal with the camera motion. Other methods Chen and Bajie (2011) and Ghosh et al. (2012) perform global motion estimation to compensate the movement of the camera and detect the objects of interest by using an edge detector. There are some proposals that approach this problem by detecting pre-defined image patches in the video sequence (Choi et al. 2013; Li et al. 2010; Yilmaz 2011). In others Hu et al. (2015), Kim et al. (2013) and Lee et al. (2010), the moving object detection is performed by building a dynamic background model to cope with the moving foreground. In Menezes et al. (2011) Markov random fields are used to model an intensity map where a spatio-temporal approach is applied to detect the objects.

In the applications where the camera is used on moving platforms the task of object tracking is also of great importance. In this scenario, however, due to the need of compensation of the camera motion, the algorithms are usually more complex (Choi et al. 2013; Kim et al. 2013; Xie et al. 2014; Yilmaz 2011).

In addition to the added complexity introduced by the use of moving cameras, if the environment to be monitored is cluttered (such as are industrial plants), the process of sorting out the useful information from the background becomes even more difficult, generally reducing the overall detection robustness. The detection of still objects with a moving camera with arbitrary trajectory is the subject of very few works in the literature. Examples can be found in Kong et al. (2010), Taneja et al. (2015), and Mukojima et al. (2016). However, due to the complex nature of this task, none of these methods is able to perform in real time.

In this paper, we propose a complete surveillance system for detecting the presence of anomalies (abandoned objects) in a cluttered environment. We use a moving camera attached to a robotic platform performing a translational movement. The monitoring system uses a reference video with no anomalies, as certified by a system operator in an initial calibration stage, similarly to the initial marking of Lee et al. (2010). The detection of anomalous objects is carried out by comparing the target video, acquired in subsequent passages of the robotic platform, with the initial reference video. All processing is performed in real time, what requires specific signal-processing solutions and makes the system suitable for a wide scope of practical situations.

The technical literature on automatic detection in surveillance applications is quite extensive. However, to the best of our knowledge, the specific problem of real-time detection of abandoned objects with a camera attached to a moving platform in a cluttered environment, such as an industrial plant, has not been fully addressed yet. Therefore, as the starting point of this work we generated a large database of surveillance videos taken from a moving camera in a cluttered industrial environment. This database, publicly available at VDAO (2014), is briefly described in Sect. 3.

A relevant issue in the type of surveillance system we are proposing is the temporal alignment of the reference and target videos. Solutions to this problem usually include external trigger signals to determine the camera position, such as a GPS device (Kong et al. 2010) or the robot's odometry (DeSouza and Kak 2002; Kundu et al. 2010). Our proposal dispenses with external signals for temporal alignment, the camera position being determined using a maximum-likelihood model for the camera movement derived directly from the acquired reference and target videos.

We also devise a multiscale approach to compare the synchronized and registered frames from the reference and target videos. In this framework, larger abandoned objects are searched in lower video resolutions and smaller objects are searched in higher resolutions, leading to an increased detection robustness at a reasonable computational cost. Video comparison includes the computation of the normalized cross-correlation (NCC) (Kong et al. 2010) between two video frames within the proposed multiresolution approach. After an NCC threshold operation, a binary detection mask is determined. Subsequent nonlinear operations, which include temporal filtering, voting step, and morphological operations, remove most false positive and false negative detections, increasing the overall system accuracy.

A step-by-step strategy for determining the system parameters was devised in order to maximize its detection rate. We describe it along with the assessment of the impact of each system variable on the resulting performance. Overall system performance is assessed using validation on a large database, recorded in a real industrial plant, comprising more than 8 h of annotated video and several types of abandoned objects(different colors, sizes, positions etc.), as detailed in da Silva et al. (2014).

It is worth mentioning that in the proposed method the detection is performed without the need of any pre-generated 3D model as those used in Taneja et al. (2015). It can also be applied without the need of selection of any Region of Interest (ROI) constraints like those applied in Kong et al. (2010), Nordlund and Uhlin (1996) and Taneja et al. (2015) and in much more complex and cluttered environments. In addition, unlike most of the previous works (Kong et al. 2010; Taneja et al. 2015), changes in the texture of the image, as well as flat objects, can be reliably detected. Considering the above, we can see that our target application, and therefore the proposed method, is unique in many ways.

To describe the proposed surveillance system, this paper is organized as follows: Sect. 2 presents the deployed system, including the video-comparison strategy in a step-by-step procedure. Section 3 briefly describes the database employed to adjust and evaluate the proposed detection scheme. Section 4 details all specific solutions developed in the context of this work to optimize the system's performance in terms of computational complexity and detection robustness. Section 5 describes the configuration of all system variables of interest, discussing their individual effects on the resulting detection process. In Sect. 6, detection results are presented characterizing the system's performance in both quantitative and qualitative ways. Finally, Sect. 7 concludes the paper emphasizing its main technical contributions.

2 General system description

The proposed surveillance system consists of a high-definition (HD) 24 frame/s camera mounted on a robotic Roomba$^{\circledR}$ platform (iRobot 2016) performing a back-and-forth movement on a horizontal track, as illustrated in Fig. 1. The moving robot takes about 3 min to cover the entire 6 m track, which oversees an industrial plant with a cluttered background.

The following framework was employed for the real-time system operation: a reference video is obtained from an initial robot passage and is validated by some operator, indicating the absence of any strange objects. The videos from all subsequent robot passages are then

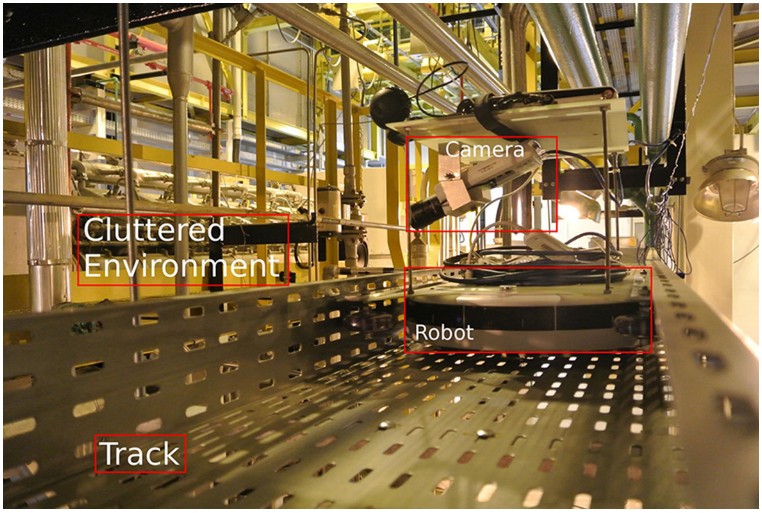


Fig. 1 Real camera-robot system with a glimpse of the cluttered environment of interest on the background

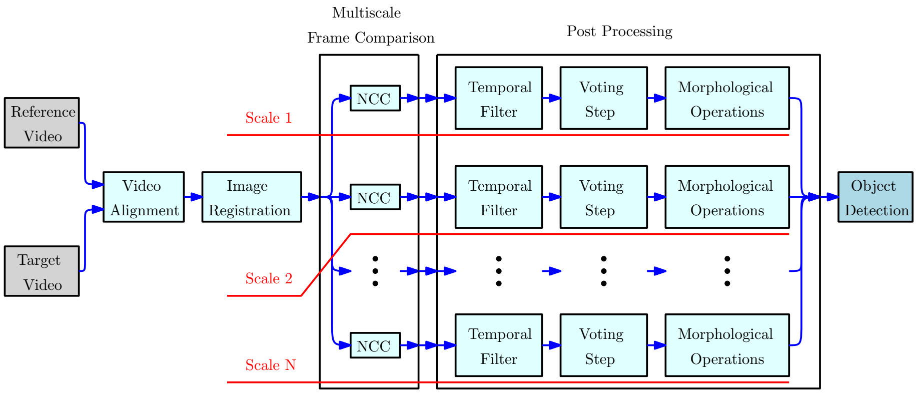


Fig. 2 Flowchart of the system. Each step is described in details in Sect. 4

compared to that reference video in search of any newly observed object. If necessary, the system operator may change the reference video using a simple update procedure, and after this the monitoring system goes back to normal operation.

A flowchart with the implemented steps of the system is shown in Fig. 2.

For a proper object detection within a given video sequence, the developed system includes the following processing steps:

(i) *Video alignment*

  (a) Reference and target video synchronization, both for the initial alignment and for correcting subsequent deviations due to small variations on the robot's speed;

(ii) *Image registration*

  (a) Identification of points of interest (PoI), usually salient points, in the corresponding reference and target frames to allow simplified real-time processing;
  (b) Geometric registration between the corresponding reference and target frames to reduce misalignments due to vibration effects on the robot movement along the rail;

(iii) *Multiscale frame comparison*

  (a) Numerically efficient (for real-time purposes) and robust frame comparison, using a multiresolution scheme, to identify significant frame discrepancies which can be associated to an abandoned object;

(iv) *Post-processing*

  (a) A pixel-level voting strategy along consecutive detection masks to identify consistent detections along time, removing occasional false detections;
  (b) Morphological operations to remove additional false-positive and false-negative situations.

Details about the techniques proposed in this paper for implementing the above stages is presented in Sect. 4.

3 Abandoned-object video database

In order to allow a systematic verification of the system robustness, a large video database, described in da Silva et al. (2014) and available at VDAO (2014), was deployed. The so-called VDAO database (from "video database of abandoned objects in a cluttered industrial environment") was recorded in a real industrial facility comprised of several machinery, pipes and other visually complex structures that pose greater challenges than those of other databases. The videos show cluttered, visually complex backgrounds and the objects suffer occlusions in the video due to the presence of structures at several different distances from the camera. The whole database comprises more than 8 hours of video and includes objects of different sizes, colors, textures and positions along the track. Also, two illumination levels (with and without additional spotlights) were addressed in the single-object videos and two different HD cameras were used to acquire the videos. There are also small differences in illumination levels in the videos because, although the videos were made indoors, they were recorded at multiple times of the day, in different days.

As a result, 8 reference videos (without abandoned objects) and 65 different target videos were produced (6 multi-object videos and 59 single object videos). The 15 objects that appear in the multi-object video set are different from the 9 objects that appear in the single-object videos. Differences in the single-object videos include illumination levels and object type and positions. The availability of videos with more than one object in the same view, together with their different sizes and types, was important to test and tune the proposed multiscale detection algorithm.

In the VDAO database, the positions of all abandoned objects within a given frame are identified in a separate file by the corresponding bounding boxes, as illustrated in Fig. 3.

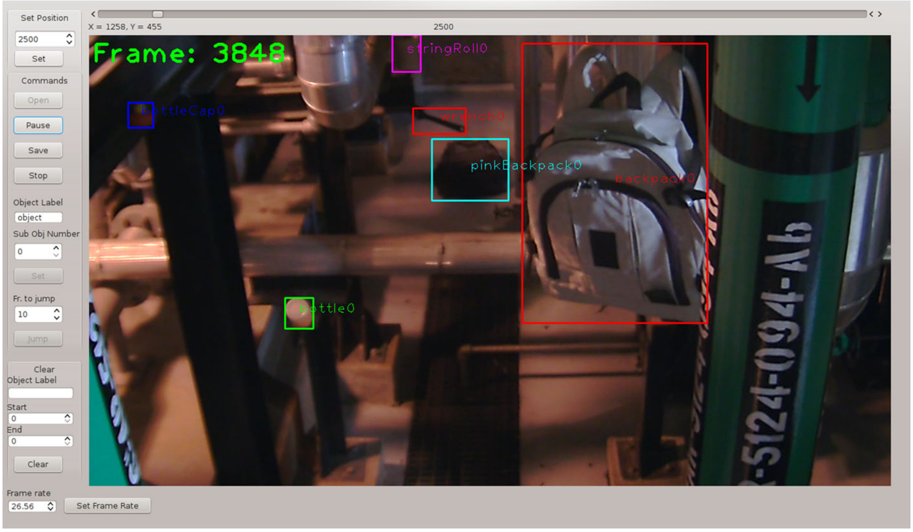

Set Position: 2500
X = 1258, Y = 455
2500
Object Label: object
Sub Obj Number: 0
Fr. to jump: 10
Start: 0
End: 0
Frame rate: 26.56

Fig. 3 Example of abandoned-object identification within the VDAO database

4 Surveillance system design

This section describes in detail all the processing steps listed in Sect. 2, including the techniques proposed in this work for a reliable, real-time operation of the monitoring system. The description that follows is based on the block diagram in Fig. 2.

4.1 Video alignment

The initial synchronization between the reference and target videos can be implemented automatically using a maximum-likelihood approach based on the video's motion data.

In that scheme, the robot's motion model assumes a constant speed along the straight track, with the direction changing when the robot reaches the track ends. The instantaneous camera speed along the track can be estimated from the homography transformation between consecutive reference and target frames, called respectively $H\{r(p-1), r(p)\}$ and $H\{t(p-1), t(p)\}$, as determined in Sect. 4.2. By integrating the horizontal component (along the track) of the camera speed, one can obtain the horizontal camera displacement in each frame index n up to a constant δ.

Figure 4 shows camera displacements as a function of the frame index estimated from actual reference (solid black curve) and target (dotted red curve) video sequences. In this plot, the maxima and minima of each curve can be associated to the two track ends. One should note that different initial positions of the robot on the track give similar curves differing only on their mean values.

The displacement curves are noisy due to the camera vibration. A noiseless motion model $d_{\mathrm{r}}(n)$, however, can be determined by performing the least-squares fitting of a piecewise-linear model composed of two straight lines of opposite angular coefficients. Such a model for the reference displacement is shown as a dashed blue line in Fig. 4, where, without any loss of generality, the direction change is assumed to be at $n = 0$. A similar motion model $d_{\mathrm{t}}(n)$ can be generated for the target displacement. Once again, since we do not know the initial position of the camera on the track, this function may have an arbitrary average level.

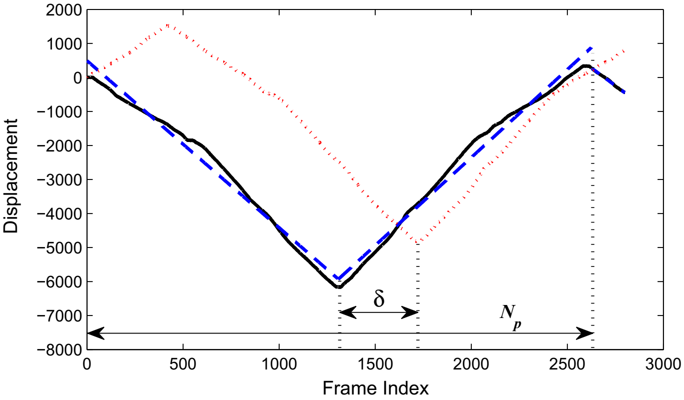


Fig. 4 Example of camera displacements estimated from reference (solid black line) and target (dotted red line) videos, where the piecewise-linear dashed blue line represents the robot's movement with a constant-speed model (Color figure online)

Using the two displacement models, an initial frame alignment between the reference and target videos can be determined as the displacement δ that maximizes the cross-covariance between the $d_{\mathrm{r}}(n)$ and $d_{\mathrm{t}}(n)$, that is

$$\hat{\delta} = \underset{\delta}{\operatorname{argmax}} \left\{ \sum_{n} (d_{\mathrm{r}}(n-\delta) - \mu_{\mathrm{r}})(d_{\mathrm{t}}(n) - \mu_{\mathrm{t}}) \right\}, \tag{1}$$

where μ_{r} and μ_{t} are the average values of $d_{\mathrm{r}}(n)$ and $d_{\mathrm{t}}(n)$, respectively.

At a first glance, the summation interval in Eq. (1) should be approximately equal to the number of frames N_p of one full back-and-forth movement of the camera. In Fig. 4, for instance, $N_p \approx 2600$ frames. However, this would impose severe restrictions for the system's real-time operation, as one would need to record a full back-and-forth cycle before synchronizing the two videos. To mitigate this issue, the summation interval can be restricted to only $\Delta \approx 200$ frames, as long as one guarantees that it contains one change in the camera moving direction to allow a proper pattern matching.

It is important to note that such initial alignment does not need to be extremely precise, since errors of a few frames translate into small displacements that can be compensated for in the image registration stage (see Sect. 4.2).

4.2 Image registration

Assuming that the reference and target videos have been initially aligned, as described in Sect. 4.1, a feature detector and descriptor algorithm was used to identify the points of interest (PoI) on two corresponding frames of both video sequences. Based on the descriptor comparisons results provided in Kucharczak et al. (2014), the SURF descriptor was chosen as it yielded a larger number of relevant keypoint correspondences in a reasonably short processing time.

In the proposed system, the HD video resolution was downsampled by a factor of 4 in each dimension, to reduce the computational complexity, allowing the system to operate in real-time.

An iterative method used to estimate parameters of a model from a set of observed data containing outliers, the random sample consensus (RANSAC) algorithm (Kong et al. 2010; Hartley and Zisserman 2003), is employed to select pairs of corresponding points in the reference and target frames. Based on these correspondences, the the homography transformation $H\{r(p), t(p)\}$ that best maps the reference PoI set onto the target PoIs is determined to compensate for any difference in the camera positioning in the temporally aligned reference and target videos (Kong et al. 2010; Hartley and Zisserman 2003). An example of such homography transformation can be seen in Fig. 5.

In this work, considering the camera's horizontal movement, all reference-target PoI correspondences yielded by the SURF algorithm with an angular displacement with absolute value larger than 1° were immediately discarded. This strategy not only reduced the computational complexity associated to the RANSAC outlier removal procedure, but also improved the consistency of the resulting homography transformation. An example of this approach is given in Fig. 6, which depicts side-by-side temporally aligned frames from the reference and target videos, along with the PoI correspondences (indicated by different color lines). From this image, one immediately observes how the angular restriction imposed on the lower plot eliminated several outlier correspondences and provided a reasonable homography transformation (represented by the green quadrilateral) for the horizontal camera movement.

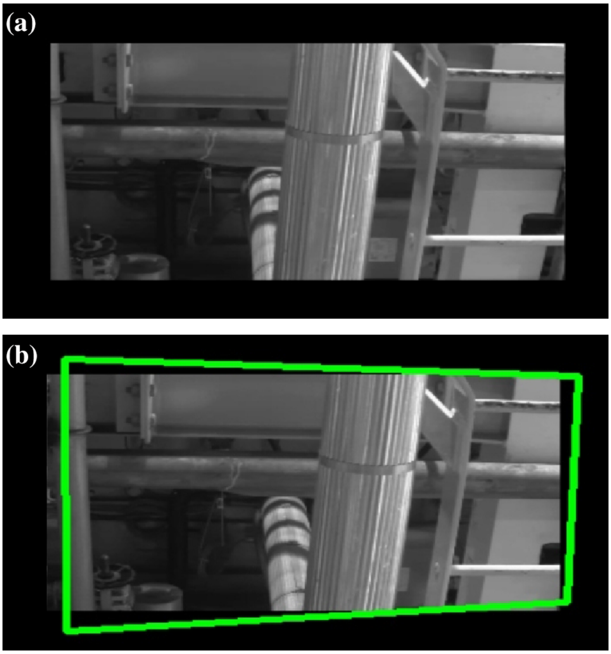


Fig. 5 Example of homography transformation. The green quadrilateral shows the image-plane mapping from the first image into the second one: **a** reference video frame; **b** target video frame (Color figure online)

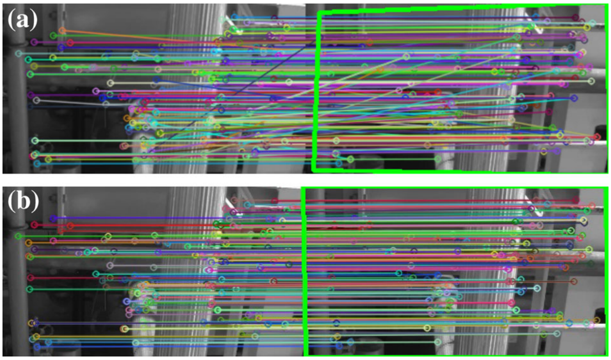


Fig. 6 Example of homographies generated with and without our angular restriction: **a** without angular restriction; **b** with angular restriction

4.3 Multiscale frame comparison

At this point, in order to perform the comparison between the corresponding aligned frames we propose the use of the normalized cross-correlation (NCC) (Kong et al. 2010) between the two images, followed by a simple threshold detection, which yields a binary image indicating areas of the target frame that are candidates to contain abandoned objects. In addition to that, in this paper we propose to apply some spatio-temporal post-processing on the binary masks generated.

The NCC $k(m, n)$ between the images $r(m, n)$ and $t(m, n)$ over a window $\mathcal{W}(m, n)$ centered in the pixel position (m, n) can be defined as

$$k(m, n) = \frac{1}{N_w} \sum_{(m', n') \in \mathcal{W}(m,n)} \frac{[r(m', n') - \bar{r}][t(m', n') - \bar{t}]}{\sigma_r \sigma_t}, \tag{2}$$

where N_w is the total number of pixels in the window $\mathcal{W}$, $\bar{r}$ and $\bar{t}$ are the average values of r and t inside the window $\mathcal{W}(m, n)$, respectively, and σ_{r} and σ_{t} are their respective standard deviations.

The NCC window size should be in the same order of the apparent size of the abandoned object to be detected, which is considered unknown or may even vary if more than one object appear on the same frame. In fact, large windows tend to overlook small objects, whereas small NCC windows may identify a single large object as several small ones. Therefore, for a robust detection, one must compute the NCC function between two frames with different window sizes, what may greatly increase the computational complexity of the resulting algorithm.

A proposed solution to this problem is to perform a multiscale NCC computation, which employs a fixed window ($K \times K$ pixels) on several downsampled versions of the reference and target videos. The whole multiscale procedure starts with a frame downsampling factor of 64, which greatly simplifies the NCC computation and makes the fixed window suitable to detect larger objects. Progressively smaller objects are then searched for with increasing resolution images.

Due to the real-time constraint, one has to restrict the allowed values of K, image resolutions, and downsampling factors. The value of K is set based on the size of the larger object to be detected in the smallest resolution to be used. For the employed database, the NCC window size was set to $K = 5$. A large K leads to missed detections due to low NCC values, as the missed wrench in Fig. 7a. On the other hand, a too small K highly increases the NCC sensitivity, yielding false-positive detections, as exemplified in Fig. 7b.

The number of scales to be employed should be decided according to the application at hand. For the VDAO database, four different image resolutions were employed, corresponding to image downsampling factors, in each direction, of 64 (suitable for the detection of larger objects), 32, 16, and 8 (smaller objects), leading to a good overall system performance, as illustrated in Fig. 8.

Figure 9 exemplifies our multiscale approach. Figure 9a shows the target image with a backpack (large object), a string roll (medium object) and a mug (small object). In Fig. 9b an image that is subsampled by 64 is used, and only the biggest object, the backpack, is detected. In Fig. 9c an image subsampled by 32 is used, and the stringroll is also detected. In Fig. 9d an image subsampled by 16 is used and the smallest object, the mug, is also detected. Note that, at this resolution, a backpack strap is also detected.

4.4 Detection mask post-processing

In order to reduce both false positive (spurious) and false negative (missed) detections, three additional processes are sequentially performed onto the resulting pixels of the NCC multiscale masks: a temporal filtering procedure, a voting strategy and opening-closing morphological operations.

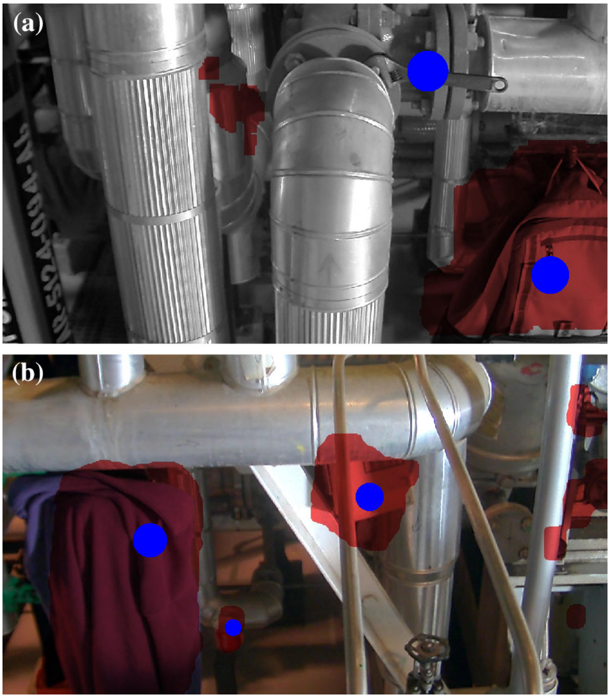


Fig. 7 Adjustment of NCC window size K (red stains indicate abandoned-object detection, and blue circles indicate a real abandoned object): **a** excessively large values of K tend to oversee smaller objects (false-negatives) such as the wrench at the top right; **b** excessively small values of K increase the sensitivity of the NCC measure, leading to false-positives, such as the regions in the lower right (Color figure online)

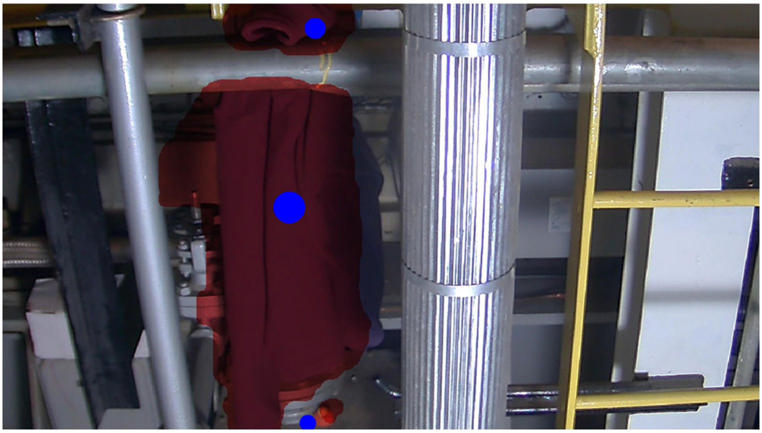


Fig. 8 Example of VDAO-database objects of different sizes being detected with the multiscale approach: The large coat is detected with the video decimation factor of 64, whereas the small bottle cap is detected with the decimation factor of 8. The blue circles indicate the abandoned objects, and the red stains, object-detection (Color figure online)

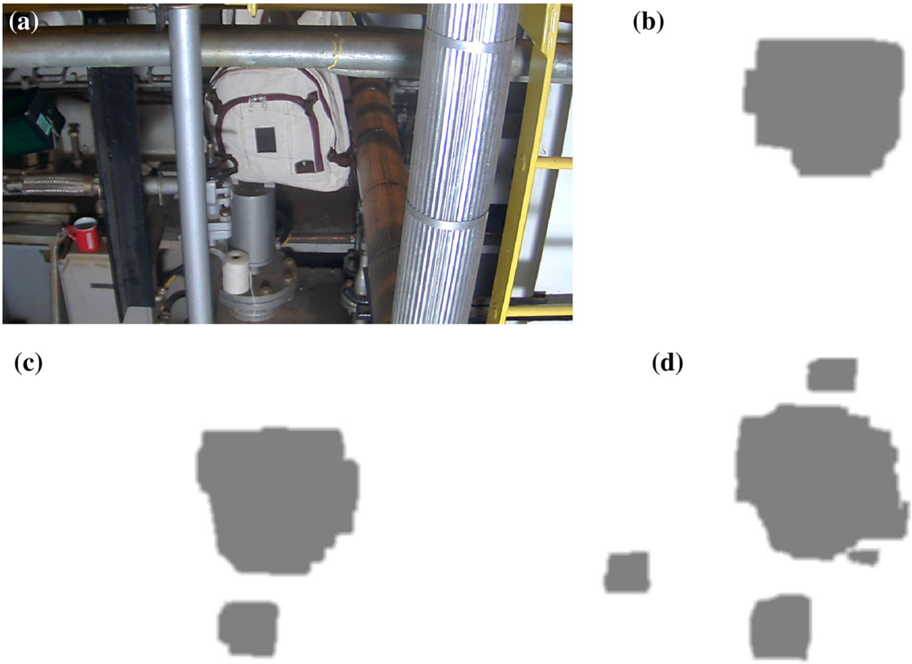


Fig. 9 Example of detection masks using the multiscale approach: in image (**b**), only the largest object (backpack) is detected; in (**c**), the string roll is also detected; finally, in (**d**), the mug algo starts to be detected. The green box is not detected in this example because it did not appear a sufficient number of times for it to be detected yet (due to the post-processing step). **a** Target image, **b** image subsampled by 64, **c** image subsampled by 32 and **d** image subsampled by 16 (Color figure online)

First, temporal filtering is applied to the NCC mask frame sequence (Kong et al. 2010) such that

$$M(m, n, p) = \prod_{i=0}^{L_{\mathrm{tf}}-1} \hat{k}(m, n, p - i), \tag{3}$$

where p is the frame index, L_{tf} is the temporal-filter length and $\hat{k}(m, n, p)$ is a binary version of the NCC output $k(m, n, p)$, defined by a threshold b_t (see Sect. 5.1). This filtering operation requires the use of the registration procedure between consecutive frames computed in Sect. 4.1, since the objects appear in different positions as the camera moves. This filtering stage is ideal to remove most of the false-positive occurrences from the detection mask, as it requires the intersection of L_{tf} consecutive masks to activate a given pixel, as seen in Fig. 10. An excessively large value of L_{tf}, however, tends to produce false negatives in our detection process. This is particularly relevant in our case since the camera is moving, and large displacements among frames that are too distant in time tend not to be dealt with properly by a homography. In our setup, the temporal filter length was set to $L_{\mathrm{tf}} = 5$. This choice will be justified in Sect. 5.2.

Following the temporal filtering, a voting procedure is employed to increase the detection robustness. The rationale for the voting procedure is that it is unlikely that an object will disappear from the video for just a few frames to reappear again later. Likewise, it is unlikely that an object will appear in the video for just a few frames. In the voting step a new mask $M_v(m, n, p)$ is generated as follows:

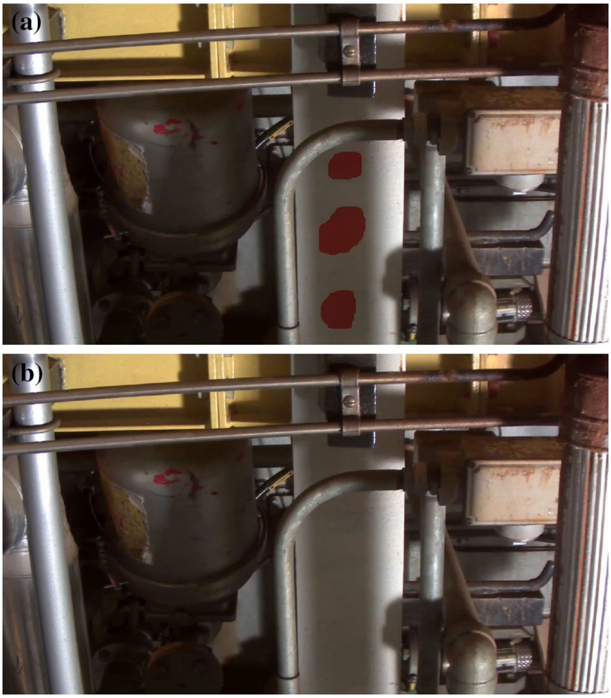


Fig. 10 Example of false-positive elimination by the temporal filtering stage: **a** detection mask (red stain) produced without temporal filtering; **b** detection mask produced with temporal filtering (Color figure online)

$$M_v(m,n,p) = \begin{cases} 1, & \text{if } \sum_{i=0}^{V-1} M(m,n,p-i) \geq v_t \\ 0, & \text{if } \sum_{i=0}^{V-1} M(m,n,p-i) < v_t \end{cases}, \tag{4}$$

where $M(m,n,p)$ is defined in Eq. (3), V is the voting interval length and v_t is the voting threshold. Once again, this procedure assumes a proper registration of the consecutive masks using a homography transformation as given in Sect. 4.2. The values of V and v_t determine the minimum amount of times a masking pixel must be activated to be recognized as part of an abandoned object. In practice, these parameters depend on the number of frames a given abandoned object appears in the target video, that is related to the camera speed. The choice of these parameters will be investigated in Sect. 5.3.

One could argue that the temporal filter [Eq. (3)] and the voting procedure [Eq. (4)] are somewhat redundant. However, there are several cases when both are necessary. Generally speaking, the temporal filter and the voting stage work in tandem to eliminate most of false positives from the detection scheme. One illustrative example, depicted in Fig. 11, is when partial occlusions generated by foreground obstacles cause an abandoned object to appear in only a limited number of frames. These situations enforce upper limit values for the voting interval length V and its corresponding threshold value v_t. Therefore, they require the temporal filter to remove a priori most of the isolated spurious detections.

In some cases, after the temporal filtering and voting steps there remains isolated small regions in the detection masks. For removing such pixels, one can perform a morphological binary opening operation. Also, there are cases when the same object is detected by more than one separate mask. In this case, the masks can be connected by a morphological binary

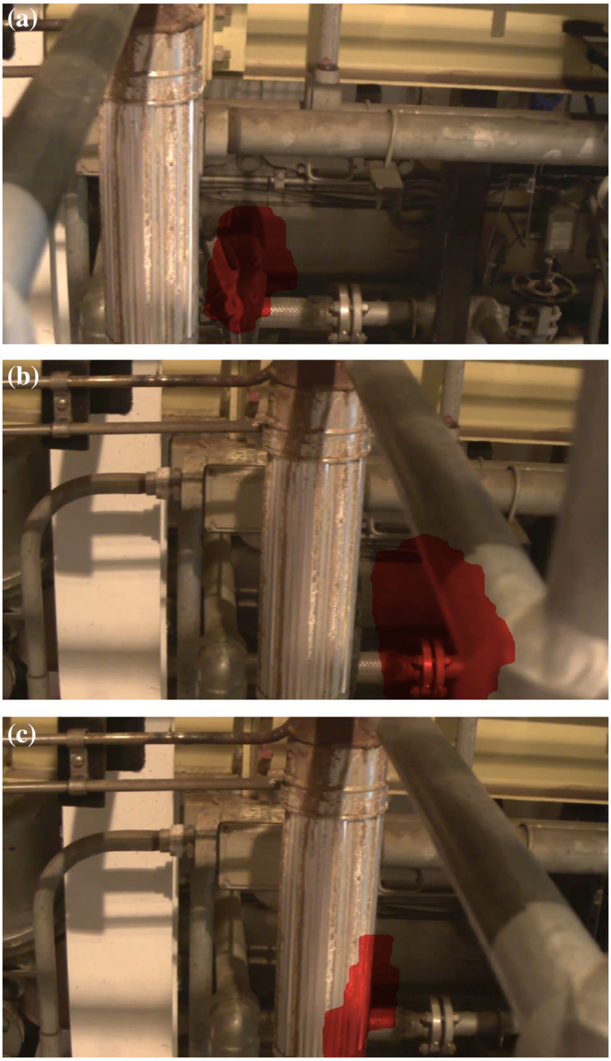


Fig. 11 Example of false positive promoted by occlusion without the temporal filtering step: **a** initial detection of abandoned object without the temporal filtering step; **b** displacement of detection mask caused by the foreground pipe in the absence of the temporal filtering (abandoned object is hidden behind bright pipe); **c** detection mask properly placed by the temporal filtering, which removes the effect from the foreground pipe

closing operation. The binary opening and closing operations can be respectively defined as (Soille 2003):

$$A \circ B = (A \ominus B) \oplus B, \tag{5}$$

$$A \bullet B = (A \oplus B) \ominus B, \tag{6}$$

where A is a binary image, B is the structuring element, $\oplus$ denotes dilation (expansion of the input image by the structuring element B), and $\ominus$ denotes erosion (contraction of the input image by B). The effect of the closing operation is illustrated in Fig. 12. In the proposed system, the closing operation is applied to the output of the opening operation.

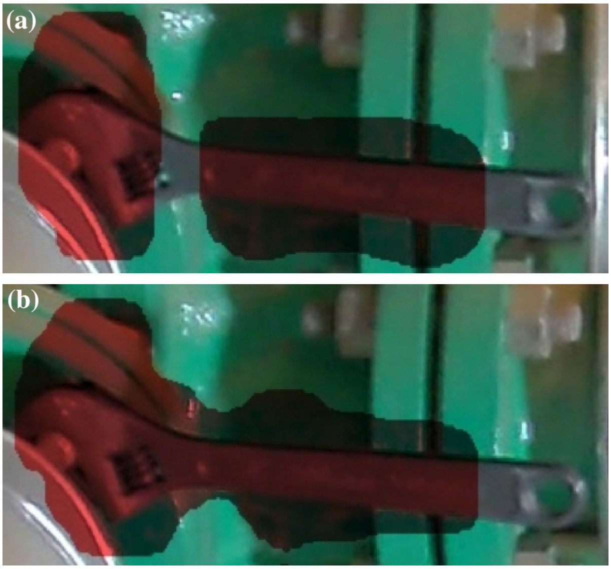


Fig. 12 Example of morphological closing (detail): **a** before; **b** after. It joins two separate masks that refer to the same object

The size of the structuring element of the opening operator should be slightly larger than the expected size of the isolated regions that may be present on the detection mask but cannot be larger than the smallest apparent size of the object intended to be detected. In contrast, for the closing operation, the structuring element size should be slightly larger than the expected size of the gaps that separate the disconnected detection masks that may be associated to the same object. In our system, we employed circular-shaped structuring elements with the same radius for the two operations. In full resolution, this radius should be approximately twice as large as the NCC window, and should keep its proportion to the resolution as it decreases.

Finally, we perform the union of the detection masks obtained in all resolutions to generate a single, final detection mask. Algorithm 1 summarizes the proposed method.

5 Tuning of the system parameters

This section outlines the design of all stages described in Sect. 4 aiming at an overall robust performance for the proposed system. As described in Sect. 3, we use the VDAO database (VDAO 2014) for this purpose. In order to do so, we divide the target VDAO sequences used in three sets. The training set comprises 16 single-object videos, while the validation set has 38 single-object videos, and the testing set is composed by 3 videos with multiple abandoned objects (those with extra illumination). The sequences on the training set are used in this section to develop routines for the adjustment of each parameter, and determine their values for the VDAO database. The effectiveness of these routines will be assessed in Sect. 6 using the validation set, and the overall algorithm performance will be assessed using the testing set.

To evaluate the effects of a given parameter in our method, several performance analysis curves are derived. In these curves, the true positive measure is determined as the percentage of the bounding box area of the abandoned objects which is covered by the detection mask,

Algorithm 1 Proposed Algorithm

```
Input: Unaligned videos $r$ and $t$
Output: Detection mask $M_o$
Parameters: $K$ (number of scales), $B$ and $C$ (morphological masks), $L_{tf}$ (temporal filtering window size) $V$ (voting window size), and $v_t$ voting threshold.
  Temporal alignment:
  Compute homographies $H\{r(p-1), r(p)\}$ and $H\{t(p-1), t(p)\}$.
  $\hat{\delta} = \text{argmax}_{\delta} \left\{\sum_n (d_{\text{r}}(n-\delta) - \mu_{\text{r}})(d_{\text{t}}(n) - \mu_{\text{t}})\right\}$,
  $\hat{t}(m, n, p) = t(m, n, p + \hat{\delta})$.
  Geometric registration:
  Extract SURF features,
  Select features with RANSAC
  Apply homography $H\{r(p), t(p)\}$ to $\hat{t} : \hat{t'} \leftarrow \hat{t}$.
  for j = 1 to $K$ do
    NCC:
    $k(m, n, p) = \frac{1}{N_w} \sum_{(m',n') \in \mathcal{W}(m,n)} \frac{[r(m',n',p) - \bar{r}][\hat{t'}(m',n',p) - \bar{t}]}{\sigma_r \sigma_t}$,
    Binarize the NCC:
    $\hat{k} \leftarrow k$;
    Temporal Filtering:
    Apply homography $H\{t(p-i), t(p)\}$ to $\hat{k}$: $\hat{k}' \leftarrow \hat{k}$
    $M_j(m, n, p) = \prod_{i=0}^{L_{\text{tf}}-1} \hat{k}'(m, n, p-i)$,
    Voting Step:
    Apply homography $H\{t(p-i), t(p)\}$ to $Mj : M'_j \leftarrow M_j$
    $M_{v(j)}(m, n, p) = \begin{cases} 1, & \text{if } \sum_{i=0}^{V-1} M'_j(m, n, p-i) \geq v_t \\ 0, & \text{if } \sum_{i=0}^{V-1} M'_j(m, n, p-i) < v_t \end{cases}$,
    Morphological Operations:
    $M_{v(j)} = (M_{v(j)} \circ B) \bullet C$,
  end for
  $M_o$ = $M_1$ or $M_2$ or $M_3$ or …or $M_K$
```

and the false positive measure is given by the percentage of remaining frame area covered by detection masks. To generate each point on the curve, we first calculate, for a given video, the average of the values obtained with each frame where the object and/or a detection spot appears. Then, we calculate the average and the standard deviation of these averages, considering all the objects in the training set. Some aspects, however, affect the performance of the proposed system, and should be taken into account by the reader when considering these measures:

- The voting operation requires a minimum number of frames to detect an abandoned object. This is usually not a problem, since the camera speed is such that there is a large number of frames between the entry of the object in the camera's field of view and its departure. However, when an object is occluded between entering and leaving the field of view, there may not be enough frames to warrant its detection, which may cause false-negative (missed) detections.
- A dual problem occurs when the voting strategy keeps the detection mask active even after the object disappears from the scene, artificially increasing the number of false-positive detections.
- As the abandoned object does not occupy its entire bounding box, there may be a significant increase on the false-negative measure.

- Abandoned objects sometimes project shadows or reflections on the other elements of the scene. Strictly speaking, since such shadows and reflections are not present in the reference video, the algorithm tends to detect them. However, the VDAO database does not consider them as objects of interest, which affect negatively the false-positive measure.

Based on all these facts, one should not expect ideal values of any pixel-based measurements, and an additional subjective validation scheme must also be employed to assess the overall system performance. However, in this parameter tuning procedure, we deemed the use of this pixel-based measurement an useful tool to, in a certain way, measure the percentage of the abandoned object covered by a detection spot, as this would help us better tune the system considering that it deals with abandoned objects of different sizes. This allows us to perform a finer parameter tuning than if we employed an object-level measurement in this procedure.

Each parameter study employs, for the other variables, the values already obtained in the previous studies.

5.1 NCC binarization threshold

The first important parameter in the proposed system is the threshold value b_t used to binarize the NCC function. Note that, from Eq. (2), its dynamic range is the interval $[-1, 1]$. In the discussion that follows, we assume that the NCC measure from Eq. (2) is normalized to be in the interval $[0, 255]$. A small value of b_t would generate too many false negatives, whereas large values of b_t would mark large numbers of abandoned-object as candidates to be processed by the system's subsequent stages. Figure 13 shows the region of convergence (ROC) curve for values of b_t in the set $\{60, 100, 140, 160, 190, 220, 250\}$. The smallest b_t corresponds to the lower leftmost point and the largest b_t to the upper rightmost point. From it, we can see that b_t can set a trade-off between true positives and false positives. Therefore, by analyzing the system's performance on a training set, an operator can control this trade-off by selecting a proper value of b_t. The performance analysis curve in Fig. 13 shows that for the training set used $b_t = 190$ is a good trade-off, with a false-positive rate of 1.4% and a true-positive rate of 53%.

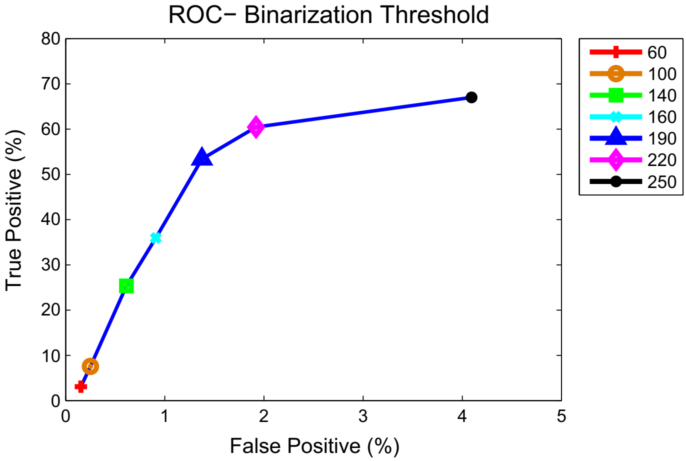


Fig. 13 ROC curve for the NCC binarization threshold variable $b_t \in \{60, 100, 140, 160, 190, 220, 250\}$

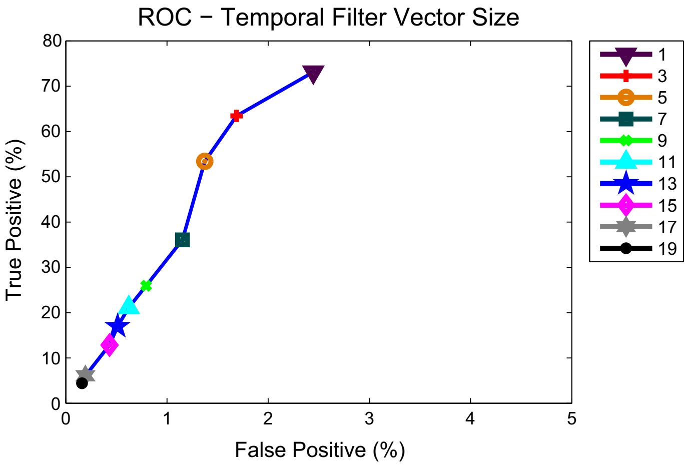


Fig. 14 ROC curve for the temporal filter length $L_{\mathrm{tf}} \in \{3, 5, 7, 9, 11, 13, 15, 17, 19\}$

5.2 Temporal filtering length

The second parameter of interest is the size of the temporal filter vector L_{tf}. As mentioned before, the temporal filtering is a necessary step to remove spurious false positives from the subsequent voting stage. Larger values of L_{tf} correspond to more restrictive temporal filters which may introduce false-negative detections.

Figure 14 shows the ROC curve associated to temporal filter vector size. From this curve, we can see that the best trade-off between true positives and false positives is given by $L_{\mathrm{tf}} = 5$. It is indeed small enough to avoid most false negatives and large enough to deal with most false positives such as the one illustrated in Fig. 11.

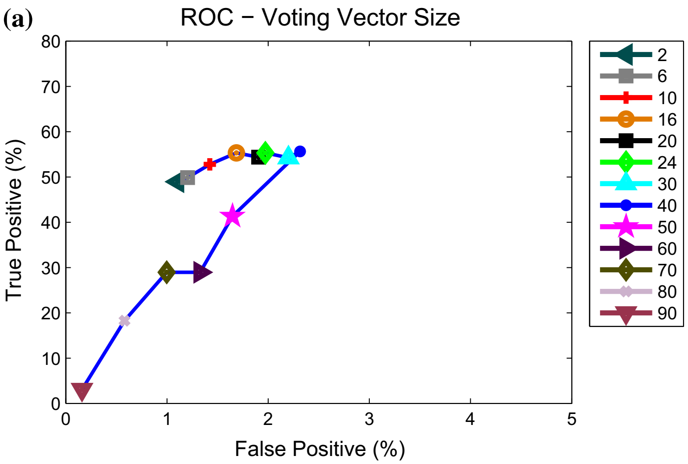
5.3 Voting parameters

The pixel-level voting procedure on the detection mask depends on two parameters, namely the length in frames V of the voting interval and the threshold value v_t, as given in Eq. (4). When testing the influence of V, v_t was set to half the value of V, and the ROC curve in Fig. 15a has been obtained for V in the set $\{2, 6, 10, 16, 20, 24, 30, 40, 50, 60, 70, 80, 90\}$. From this curve, for $V \geq 50$, the true positive rates decrease without any improvement upon the false positive rates. This result is to be expected because, as the sizes of V and v_t increase, the number of objects that would not appear in a sufficient number of frames in order for them to be detected would also increase.

The best trade-offs are delivered with $V = 16$, which are then considered in the subsequent analysis on the value of v_t.

Figure 15b shows the results for $V = 16$ and $v_t \in \{1, 4, 7, 10, 13, 16\}$. From the ROC curves obtained from the study of the voting parameters, the best compromise is given by $V = 16$ and $v_t = 7$ frames, with a false positive rate of 1.83% and a true positive rate of 56.60%.

Table 1 summarizes the values obtained for each variable studied in this section:



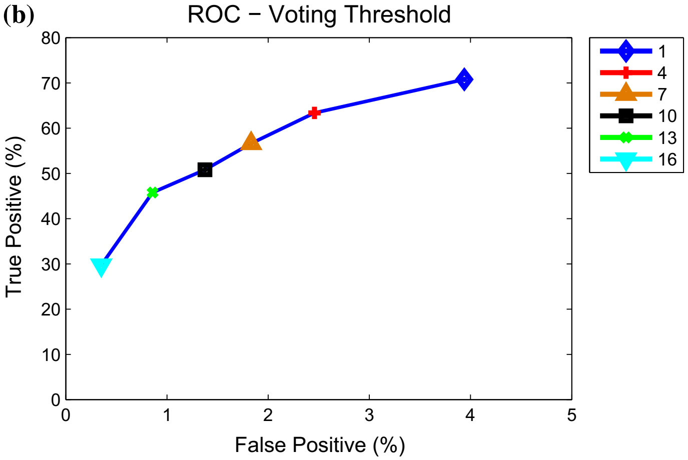


Fig. 15 ROC curves for the voting-stage parameters: **a** length in frames V of the voting interval; **b** threshold value v_t

Table 1 Table showing the values obtained for the studied variables

Variable name	Variable	Value
NCC binarization threshold	b_t	190
Temporal filtering length	L_{tf}	5
Voting vector length	V	16
Voting threshold	v_t	7

6 Experimental results

In this section, we assess the parameter tuning procedure presented in Sect. 5 and the overall system performance using the remaining database videos not employed in the parameter tuning stage.

Table 2 Comparison between the detection results obtained with the training and validation sets

Data employed	True positive		False positive	
	μ (%)	σ (%)	μ (%)	σ (%)
Training set	57	21	1.8	1.3
Validation set	66	15	3.6	3.6

6.1 Validation results with the VDAO database

Initially, we have computed the true positive and false positive rates (as defined in Sect. 5) for the validation-set videos. The frame-based detection results (average (μ) and standard deviation (σ) values) are shown in Table 2 for both the training and validation sets. From these numbers, on can conclude that the parameter tuning detailed in Sect. 5 yields similar and consistent results with both video sets, indicating good generalization capability for the resulting system. It is important to note that despite the not so high values of true positive rates (of the order of 60%), these refer to the percentage of the bounding boxes covered in a pixel-by-pixel level, as specified in Sect. 5. However, in a practical application, an operator will see the masks and decide whether an abandoned object has been detected or not. We have performed this evaluation in the validation set, and verified that all the abandoned objects contained in the validation dataset were properly detected.

6.2 Additional performance metrics

In order to assess the overall system performance, we employed a testing set consisting of the 3 multi-object videos from the VDAO database (VDAO 2014), with 15 abandoned objects in each video. This multi-object scenario requires more complex metrics than before, as in principle one does not know beforehand which ground-truth frame mask corresponds to the mask of a given detected object. In addition, in cases of missed or false detections, there is no one-to-one correspondence with the ground truth masks. Due to these matters, the initial system evaluation was based on Nawaz et al. (2014), which proposes metrics that take into account the accuracy of the matchings as well as the false positives and false negatives for the case of multiple objects. In addition, we also assess the proposed method using metrics that mimic whether or not an operator, by looking at the detection masks generated, is able to indicate correctly the presence of an abandoned object.

The definition of the metrics in Nawaz et al. (2014) is based on two pixel sets:

- $A_{k,i}$, the set of pixel positions belonging to the ground truth mask of the object of index i in frame k.
- $\hat{A}_{k,i}$, the set of pixel positions belonging to the bounding box of the detected mask of the object of index i in frame k.

The first metric employed here is the accuracy error $\mathcal{A}_k$, that represents the mismatch extent between the estimated and ground-truth states at frame k. This metric has to be computed over all possible detected-objects and ground-truth mask correspondence. Hence, by defining

$$\mathcal{O}(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{7}$$

where $|A|$ is the number of elements of the set A, then the accuracy error is given by

$$\mathcal{A}_k = \min_{\pi \in \Pi_{\max(d_k, \hat{d}_k)}} \sum_{i=1}^{\min(d_k, \hat{d}_k)} [1 - \mathcal{O}(\hat{A}_{k,i}, A_{k,\pi(i)})], \tag{8}$$

where d_k and $\hat{d}_k$ are the numbers of ground-truth masks and detected objects in frame k, respectively, the permutation $\pi(i)$ is a one-to-one function that maps the set of indexes of detected objects into the indexes of ground truth masks, and Π_j is the set of all possible permutations over j indexes. If $d_k > \hat{d}_k$, then $j = d_k$, otherwise $j = \hat{d}_k$. In Eq. (8), $\mathcal{O}(\hat{A}_{k,i}, A_{k,\pi(i)}) \in [0, 1]$ represents the amount of spatial overlap between the detected-object bounding box $\hat{A}_{k,i}$ and its corresponding ground-truth value. Essentially, Eq. (8) determined all possible spatial overlaps between the detected and ground-truth bounding boxes in frame k and selects the combination that minimizes the mismatch A_k between the estimated and ground-truth states.

Another metric is frame-level accuracy error rate (AER) defined as the average of $\mathcal{A}_k$ over all frames, that is

$$\text{AER} = \frac{1}{K} \sum_{k=1}^{K} \mathcal{A}_k, \tag{9}$$

where K is the total number of frames. The minimum accuracy error happens when all the bounding boxes coincide exactly pixel by pixel, and is therefore zero. The maximum value happens when no bounding boxes coincide, and is equal to $\min(d_k, \hat{d}_k)$.

The problem with $\mathcal{A}_k$ is that if an object is missed or there is a false positive, its contribution to $\mathcal{A}_k$ is zero. To cope with that issue, the authors of Nawaz et al. (2014) define the cardinality error rate (CER), that quantifies the discrepancy in estimating the number of targets. It is just the average difference in the numbers of ground truth masks and detected objects over all frames, that is

$$\text{CER} = \frac{1}{K} \sum_{k=1}^{K} \mathcal{C}_k. \tag{10}$$

where $\mathcal{C}_k = |d_k - \hat{d}_k|$.

Therefore, a metric that would take into account both the accuracy and the effect of false positives and false negatives can be obtained by adding both $\mathcal{A}_k$ and $\mathcal{C}_k$. In addition, since both increase with the number of objects present, this sum can be normalized by the number of objects to produce the multiple extended-target tracking error (METE) score METE_k defined as (Nawaz et al. 2014)

$$\text{METE}_k = \frac{\mathcal{A}_k + \mathcal{C}_k}{\max(d_k, \hat{d}_k)}. \tag{11}$$

In Nawaz et al. (2014) it is shown that the METE value is equal to 0 in the case of perfect matching and is 1 in the worst case.

6.3 Testing results with the VDAO database

Table 3 shows the AER, CER and METE results, averaged over all frames, for the tuned system using the testing video set. The high METE scores, around 78% for each video, indicate discrepancies between the detected-object and ground-truth bounding boxes, what is easily understood from the discussion provided in Sect. 5, as illustrated in Fig. 16. In fact, in such a case, only about 50% of the bounding box for the string-roll detection mask, in

Table 3 Quantitative evaluation of object-detection system's performance using metrics proposed in Nawaz et al. (2014)

Data employed	AER	CER	METE	
	μ	μ	μ	σ
Multi-object video 1	2.5	0.9	0.72	0.11
Multi-object video 2	2.5	2.2	0.77	0.14
Multi-object video 3	3.2	2.9	0.85	0.09

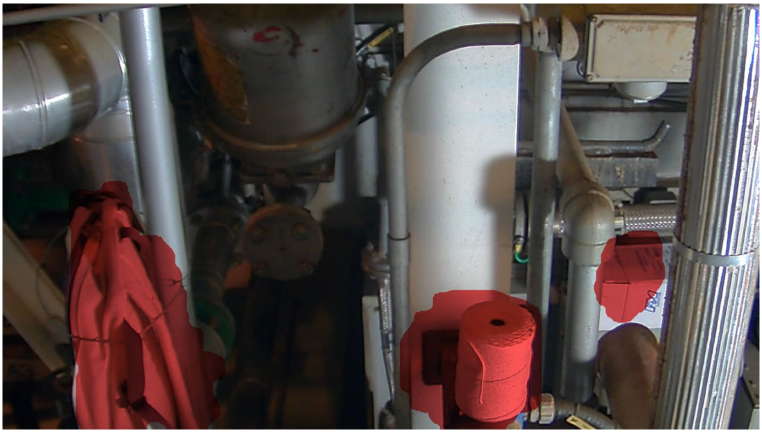


Fig. 16 Detection results for frame with objects of different sizes (backpack on the left, box on the right and string roll in the middle). All of them have been correctly detected

the middle bottom of the frame, coincides with its ground truth, justifying the large METE values attained.

Despite the large METE values yielded by the proposed system, an operator can correctly indicate the presence of the string roll by looking at the provided detection mask. The same is true for the backpack and the box objects, also shown in that frame image, and the object-based detection error rate should be zero in that case. This demonstrates the importance of measuring the detection error rate at the object level, which can be done based on the following measurements:

- The number of true positives N_{TP_j}, that is the number of frames where object j is correctly detected.
- The overall number of false positives $N_{\mathrm{FP}_{\mathrm{all}}}$, that is the number of frames where any object is incorrectly indicated.
- The number of false negatives N_{FN_j}, that is the number of frames where object j is missed.
- The overall number of true negatives $N_{\mathrm{TN}_{\mathrm{all}}}$, that is the number of frames that have been correctly indicated as having no objects.
- The number of positives N_{P_j}, that is the number of frames where object j is present.
- The overall number of positives N_{P}, that is the number of frames where the presence of any object is indicated.
- The number of negatives N_{N_j}, that is the number of frames where object j is not present.
- The overall number of negatives N_{N}, that is the number of frames where no object is indicated.

Table 4 Quantitative evaluation of object-detection system's performance in terms of the correct detection of an abandoned object [Eqs. (12)–(15)]

Data employed	TP	FP	TN	FN
Multiobject video 1	0.68	0.40	–	0.13
Multiobject video 2	0.69	0.42	–	0.96
Multiobject video 3	0.77	0.43	–	0.08
Average	0.71	0.42	–	0.10

Note that the number of true negatives cannot be computed for a particular object j, since the frames that have been correctly classified as having no object cannot be associated with any object. In addition, one cannot assign a false positive to any object. Then, using the above measurements the following detection error rate metrics are defined:

$$\text{TP} = \frac{\sum_{j=1}^{\text{N}_{\text{objs}}} N_{\text{TP}_j}}{\sum_{j=1}^{\text{N}_{\text{objs}}} N_{\text{P}_j}}, \tag{12}$$

$$\text{FP} = \frac{N_{\text{FP}_{\text{all}}}}{N_{\text{P}}}, \tag{13}$$

$$\text{TN} = \frac{N_{\text{TN}_{\text{all}}}}{N_{\text{N}}}, \tag{14}$$

$$\text{FN} = \frac{\sum_{j=1}^{\text{N}_{\text{objs}}} N_{\text{FN}_j}}{\sum_{j=1}^{\text{N}_{\text{objs}}} N_{\text{N}_j}}, \tag{15}$$

where K is the total number of frames and N_{objs} is the total number of objects.

Table 4 shows the resulting values for these metrics using our multi-object testing set. The average number of true positives as defined by Eq. (12) is approximately 71%, and the average number of false positives [Eq. (13)] is about 42%, most of them due to the object shadows and reflections, which are multiplied in the cluttered environment, as already discussed in Sect. 5. In addition, one can see that the average value of false negatives [Eq. (15)] is approximately 10.0%, meaning that there are not many missed frames containing abandoned objects. This indicates that the proposed abandoned object detection system can be useful for providing effective alarms of the presence of abandoned objects in a practical situation. Note that the void entries in Table 4 are accounted for the fact that, in the particular case of the multi-object videos of the VDAO databases, there are no frames without any abandoned objects. Thus, both $N_{\text{TN}_{\text{all}}}$ and N_{N} are zero, and TN is undefined.

The proposed metrics given in Eqs. (12)–(15) are relevant for the assessment of the overall system capabilities and its intrinsic behaviors. In a practical surveillance scenario, however, one is mostly interested in a system that can give an alarm for all abandoned objects. If the alarm happens in all frames in which the object is present or just in some, the practical effect is the same, drawing the attention of an operator that can further analyze the surveillance video. We have performed this analysis on the three videos from the testing set, and have verified that the proposed system has alarmed all 15 objects in the 3 videos. This suggests the usefulness of the proposed method in a practical surveillance scenario.

As far as complexity is concerned, for real-time applications one has to be able to run the detection algorithm in the time interval between two frames. If this time interval is not large enough due to processing power restrictions, one has to subsample the videos temporally. The problem with that solution is that if a video is subsampled by a very large factor, there may be not enough frames where an object is present in order for us to perform the temporal filtering and voting procedures [Eqs. (3), (4)]. Furthermore, to alleviate the negative effects of the temporal subsampling of the videos, we can reduce the robot speed and then subsample the videos. In this case there may be both enough time for processing between frames and an adequate number of frames for the temporal filtering and voting.

In our experiments, we employed an Intel core I7 2630QM processor with a 2-GHz clock rate, and with 8 GB of RAM. This allowed real-time operation with a temporal subsampling of 8 (3 frames/s), which was enough for the algorithm to work at the robot speed as given in Sect. 2.

6.4 Comparison with state-of-the-art methods

In Kong et al. (2010), the detection of abandoned objects with a moving camera (DAOMC) mounted on a car explores several particular scenario characteristics, such as: (i) target regions are constrained to road sides; (ii) all processed images contain horizon lines; (iii) parallax is avoided by dealing with far enough objects; (iv) an available GPS signal is used to synchronize the reference and target videos.

With a somewhat similar goal, in Mukojima et al. (2016) a camera is mounted in the frontal part of a train to detect anomalies such as abandoned objects along the rails. The proposed moving-camera background subtraction (MCBS) method employs an optical flow algorithm (Weinzaepfel et al. 2013) to perform geometric registration. There, the regions of interest are only the train rails, which exert a similar role as that of the road in Kong et al. (2010). By limiting the area of the image when searching for anomalies, an excessive amount of false positives is avoided in both works.

In Nakahata et al. (2017), a two-stage dictionary learning process is used in a spatio-temporal composition for moving-camera detection (STC-mc) of anomalies between two video sequences. In that approach, portions of the target video that are poorly represented by the dictionary are considered anomalies. This method does not employ motion estimation, tracking, background subtraction, temporal alignment or geometric alignment.

In the present contribution, we compare the proposed method, hereby referred to as the anomaly detector using multiscales (ADMULT) algorithm, to the DAOMC (Kong et al. 2010), MCBS (Mukojima et al. 2016) and STC-mc (Nakahata et al. 2017) systems. In this comparison, the post-processing steps to find the rails in the MCBS algorithm were removed to make it fit to a more general surveillance scenario. In the same way, the restriction to detect anomalies only under the horizon line was removed from the DAOMC approach. Also, for this algorithm, the reference and target video synchronization was performed manually, since the GPS signal was not available.

Following the methodology employed in Nakahata et al. (2017), the metrics employed to evaluate the methods were the following: true positives occur when at least one pixel of the detection mask falls into the abandoned object bounding box, used as the ground truth; false positives occur when there is no intersection between the detection mask and an abandoned object bounding box; false negatives occur when no pixel from a detection mask falls inside an abandoned object bounding box; and true negatives occur when there are neither abandoned objects nor detection masks in the given frame. This analysis is made frame by frame, and after all the frames from a given video clip are taken into account, an average value is calculated for the metric. We also consider the minimum distance (DIS) of all operating points to the ideal point (TP = 1 and FP = 0) in the TP × FP plane, that is

$$DIS = \sqrt{(1 - \mathrm{TP})^2 + \mathrm{FP}^2}. \tag{16}$$

Table 5 Comparison results of the proposed ADMULT method, STC-mc, DAOMC, and MCBS considering the same seven video clips employed in Nakahata et al. (2017)

Object	STC-mc			DAOMC			MCBS			ADMULT		
	TP	FP	*DIS*	TP	FP	*DIS*	TP	FP	*DIS*	TP	FP	*DIS*
Dark blue box 1	1.00	0.04	0.04	**1.00**	**0.00**	**0.00**	1.00	0.90	0.90	**1.00**	**0.00**	**0.00**
Towel	0.92	0.01	0.08	1.00	0.10	0.10	1.00	0.07	0.07	**1.00**	**0.00**	**0.00**
Shoe	0.90	0.04	0.11	1.00	0.04	0.04	1.00	0.28	0.28	**1.00**	**0.00**	**0.00**
Pink bottle	0.99	0.13	0.13	1.00	1.00	1.00	1.00	0.96	0.96	**1.00**	**0.00**	**0.00**
Camera box	1.00	0.03	0.03	**1.00**	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**
Dark blue box 2	0.37	0.42	0.76	1.00	1.00	1.00	1.00	0.10	0.10	**1.00**	**0.00**	**0.00**
White jar	0.29	0.64	0.96	1.00	0.10	0.10	1.00	0.99	0.99	**1.00**	**0.00**	**0.00**
Average	0.78	0.19	0.59	1.00	0.32	0.32	1.00	0.47	0.47	**1.00**	**0.00**	**0.00**

The best results are displayed in bold

In an initial experiment, only the same seven 200-frame VDAO video excerpts employed in Nakahata et al. (2017) were considered. In these video portions, all target frames contain an abandoned object. Therefore, only the TP, FP and DIS metrics are shown in Table 5, where one can observe the superior performance achieved by the proposed ADMULT method.

For a more comprehensive performance assessment, detection experiments were devised with 200-frame excerpts from all the 59 single-object VDAO videos, as made available in VDAO-200 (2017). It is important to note that all methods were configured using only the seven videos employed in Nakahata et al. (2017), whose results are listed in Table 5. In the broader 59-video dataset, however, there are many occlusions, objects entering or leaving the scene and frames without any abandoned object. For example almost half (27 out of 59) of the selected videos present some type of occlusion, while at least five of the videos present objects that are completely occluded part of the time. Also in at least 21 of the selected excerpts the objects are partially or completely covered by shadows cast by the environment. In addition, there are more variations in illumination, in object shapes and in the amount of camera vibration, which have a negative effect on the algorithm performances. Results for each video in this increased dataset are shown in Table 6, whereas Table 7 summarizes the average performances of all algorithms for the same dataset. Such results once again indicate a superior overall performance of the ADMULT system (as given by the *DIS* measurement), despite a lower TP value, which is compensated by the considerably better FP and TN scores.

6.5 Processing-time experiment

In a final experiment, we evaluate the processing time of all discussed methods for processing all 200 frames in the seven videos analyzed in Table 5. Results are shown in Table 8, considering an Intel i7-4790K CPU with 4.0 GHz and 32 GB of RAM. From such results, one clearly observes how the proposed ADMULT method is much faster than all its competitors, being more than twice as fast as any of the others.

Table 6 Comparison results of the proposed ADMULT method, STC-mc, DAOMC, and MCBS considering the whole VDAO-200 database

Object	STC-mc			DAOMC			MCBS			ADMULT		
	TP	FP	*DIS*	TP	FP	*DIS*	TP	FP	*DIS*	TP	FP	*DIS*
Object 1	0.37	0.42	0.76	1.00	1.00	1.00	**1.00**	**0.10**	**0.10**	1.00	0.63	0.63
Object 2	1.00	0.04	0.04	**1.00**	**0.00**	**0.00**	1.00	0.90	0.90	**1.00**	**0.00**	**0.00**
Object 3	0.90	0.04	0.11	1.00	0.04	0.04	1.00	0.28	0.28	**1.00**	**0.00**	**0.00**
Object 4	1.00	0.03	0.03	**1.00**	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**
Object 5	0.92	0.01	0.08	1.00	0.10	0.10	**1.00**	**0.07**	**0.07**	0.71	0.95	0.95
Object 6	0.29	0.64	0.96	1.00	0.10	0.10	1.00	0.99	0.99	**1.00**	**0.00**	**0.00**
Object 7	0.99	0.13	0.13	1.00	1.00	1.00	1.00	0.96	0.96	**1.00**	**0.00**	**0.00**
Object 8	0.00	0.01	1.00	1.00	0.87	0.87	**0.75**	**0.31**	**0.39**	0.54	0.02	0.47
Object 9	0.00	1.00	1.41	0.94	1.00	1.00	**0.67**	**0.18**	**0.37**	0.52	0.06	0.48
Object 10	0.01	0.01	0.99	1.00	0.97	0.97	**0.89**	**0.10**	**0.15**	0.69	0.00	0.31
Object 11	0.03	0.79	1.25	0.98	0.98	0.98	**0.73**	**0.32**	**0.42**	0.67	1.00	1.05
Object 12	0.20	0.07	0.81	0.94	0.48	0.48	0.87	1.00	1.01	**1.00**	**0.22**	**0.22**
Object 13	0.00	0.50	1.12	0.86	0.71	0.72	**0.84**	**0.00**	**0.16**	0.64	0.19	0.40
Object 14	0.08	0.05	0.92	1.00	0.74	0.74	**0.92**	**0.01**	**0.08**	1.00	0.15	0.15
Object 15	0.00	1.00	1.41	1.00	1.00	1.00	0.89	1.00	1.01	**0.59**	**0.04**	**0.42**
Object 16	0.00	0.08	1.00	0.77	1.00	1.02	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**	**1.00**
Object 17	0.06	1.00	1.37	0.96	0.46	0.46	**0.80**	**0.12**	**0.23**	0.62	0.30	0.48
Object 18	0.00	0.09	1.00	0.75	0.99	1.02	**0.43**	**0.00**	**0.57**	0.00	0.23	1.03
Object 19	0.00	0.03	1.00	1.00	0.67	0.67	**0.89**	**0.00**	**0.11**	0.54	0.15	0.48
Object 20	**0.36**	**0.50**	**0.81**	0.26	1.00	1.24	0.67	1.00	1.05	0.00	0.00	1.00
Object 21	0.00	0.68	1.21	0.97	0.62	0.62	**0.95**	**0.61**	**0.61**	0.97	0.72	0.72
Object 22	0.00	0.07	1.00	1.00	0.90	0.90	**0.92**	**0.05**	**0.09**	0.68	0.75	0.81
Object 23	0.00	0.83	1.30	0.93	1.00	1.00	0.97	1.00	1.00	**1.00**	**1.00**	**1.00**
Object 24	0.58	0.93	1.02	0.00	1.00	1.41	0.00	0.73	1.24	**0.00**	**0.00**	**1.00**
Object 25	0.00	0.02	1.00	1.00	0.90	0.90	**0.58**	**0.00**	**0.43**	0.56	0.55	1.00
Object 26	0.00	0.06	1.00	1.00	0.54	0.54	**0.87**	**0.05**	**0.14**	0.64	0.01	0.70
Object 27	0.26	0.34	0.82	1.00	0.72	0.72	1.00	1.00	1.00	**1.00**	**0.10**	**0.36**
Object 28	0.01	0.01	1.00	1.00	0.89	0.89	**1.00**	**0.00**	**0.00**	1.00	0.00	0.10
Object 29	0.00	0.14	1.01	0.91	0.98	0.98	**0.76**	**0.02**	**0.24**	0.68	0.01	0.32
Object 30	0.00	0.01	1.00	1.00	0.97	0.97	0.80	0.49	0.53	**0.56**	**0.00**	**0.44**
Object 31	0.00	0.01	1.00	**1.00**	**0.61**	**0.61**	0.87	0.80	0.81	0.61	0.55	0.67
Object 32	0.00	0.01	1.00	1.00	0.78	0.78	**0.83**	**0.00**	**0.17**	0.32	0.00	0.68
Object 33	0.78	0.81	0.83	0.83	1.00	1.01	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Object 34	0.00	0.02	1.00	1.00	0.69	0.69	**0.70**	**0.00**	**0.30**	0.56	0.00	0.44
Object 35	0.00	0.97	1.39	0.97	0.62	0.62	0.87	0.82	0.83	**0.62**	**0.01**	**0.38**
Object 36	0.24	1.00	1.26	0.02	1.00	1.40	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Object 37	0.43	0.18	0.59	0.99	1.00	0.96	**0.93**	**0.00**	**0.07**	**0.93**	**0.00**	**0.07**
Object 38	0.00	1.00	1.41	1.00	0.99	0.99	**0.71**	**0.05**	**0.30**	0.44	0.00	0.56
Object 39	0.09	0.04	0.91	0.91	1.00	1.00	0.84	0.93	0.94	**1.00**	**0.25**	**0.25**
Object 40	0.56	0.44	0.92	1.00	0.95	0.95	1.00	0.56	0.56	**1.00**	**0.14**	**0.14**
Object 41	0.00	0.78	1.27	0.64	0.99	1.05	**0.88**	**0.87**	**0.87**	0.87	1.00	1.01

Table 6 continued

Object	STC-mc			DAOMC			MCBS			ADMULT		
	TP	FP	*DIS*	TP	FP	*DIS*	TP	FP	*DIS*	TP	FP	*DIS*
Object 42	0.00	1.00	1.41	0.96	0.96	0.96	0.88	0.91	0.91	**0.49**	**0.00**	**0.51**
Object 43	0.00	0.08	1.00	0.72	1.00	1.04	**0.14**	**0.00**	**0.86**	0.00	0.00	1.00
Object 44	0.00	0.19	1.02	0.96	1.00	1.00	**0.73**	**0.14**	**0.31**	0.63	0.00	0.37
Object 45	0.15	0.92	1.25	0.01	1.00	1.41	0.82	1.00	1.02	**1.00**	**1.00**	**1.00**
Object 46	0.00	0.43	1.09	0.93	0.97	0.97	0.95	0.79	0.79	**0.99**	**0.14**	**0.14**
Object 47	0.01	0.20	1.01	1.00	1.00	1.00	**0.93**	**0.00**	**0.07**	0.91	0.22	0.24
Object 48	0.00	0.01	1.00	0.96	0.97	0.97	**0.72**	**0.16**	**0.32**	0.42	0.00	0.58
Object 49	0.00	0.04	1.00	1.00	0.99	0.99	**1.00**	**0.06**	**0.06**	0.93	0.00	0.07
Object 50	0.00	0.02	1.00	1.00	0.77	0.77	**0.86**	**0.14**	**0.20**	0.18	0.89	1.21
Object 51	0.01	0.86	1.31	0.97	0.92	0.92	**0.85**	**0.66**	**0.68**	1.00	1.00	1.00
Object 52	0.00	0.68	1.21	0.40	1.00	1.17	**0.63**	**0.79**	**0.87**	0.84	1.00	1.01
Object 53	0.06	0.82	1.25	0.79	1.00	1.02	0.69	1.00	1.05	**0.88**	**1.00**	**1.01**
Object 54	0.00	0.20	1.02	1.00	0.51	0.51	**0.84**	**0.01**	**0.16**	0.50	0.00	0.50
Object 55	0.39	0.75	0.96	0.86	1.00	1.01	0.59	0.32	0.52	**0.49**	**0.00**	**0.51**
Object 56	0.52	0.45	0.65	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	**0.38**	**0.38**
Object 57	0.36	0.09	0.65	0.96	0.92	0.92	1.00	0.67	0.67	**1.00**	**0.21**	**0.21**
Object 58	0.00	0.05	1.00	0.97	0.80	0.80	**0.62**	**0.00**	**0.38**	0.18	0.00	0.82
Object 59	0.00	1.00	1.41	1.00	1.00	1.00	0.73	0.79	0.83	**0.53**	**0.00**	**0.47**

The best results are displayed in bold

Table 7 Average results for the detection methods ADMULT, STC-mc, DAOMC and MCBS systems considering all 59 single-object videos of the VDAO database

Method	TP	FP	TN	FN	DIS
STC-mc	0.19	0.42	0.58	0.81	0.91
DAOMC	0.83	0.43	0.54	0.17	0.46
MCBS	**0.89**	0.84	0.02	**0.11**	0.85
ADMULT	0.71	**0.28**	**0.63**	0.29	**0.40**

The best results are displayed in bold

Table 8 Required processing time (in seconds) for each detection method when processing each of the seven 200-frame videos of the VDAO database. The hardware platform used was an Intel i7-4790K CPU with 4.0 GHz and 32 GB of RAM

Object	STC-mc	DAOMC	MCBS	ADMULT
Dark blue box 1	433	265	50924	106
Towel	345	280	50403	105
Shoe	542	293	50427	112
Pink bottle	415	280	50170	121
Camera box	448	299	50238	115
Dark blue box 2	221	289	51740	114
White jar	248	282	49901	128
Average	378	284	50543	114

7 Conclusions and future work

In this paper, a new technique for real-time detection of abandoned objects in a cluttered environment was described. Among the many innovations incorporated onto the proposed system, we may highlight: a maximum-likelihood video-alignment approach that precludes the use of any external trigger signals; a multiresolution video analysis that speeds up the overall process and allows one to detect multiple abandoned objects of different apparent sizes, in each video frame; a temporal-filtering procedure that improves the time consistency of the abandoned detection process, thus eliminating false-positive detections; use of morphological operations to remove isolated small detections and to connect close-by detection spots.

Another contribution of this work was the description of a complete procedure for tuning the algorithm parameters, the effectiveness of which was evaluated using an independent validation video set. The VDAO database used in this work comprises more than 8 hours of recorded and annotated video in a real industrial environment and has been made publicly available at VDAO (2014). We have performed the overall system evaluation employing a large set of testing videos and several distinct metrics (including processing time) to assess the system performance. Extensive results show that the proposed method operates much faster than the other state-of-the-art methods, running at least 2.5 times faster than every other compared method in the performed experiments. Our experiments have shown that the proposed method is able to detect abandoned objects in real time with good true positive rates and low false negative rates. In one of the experiments our method has proven to perform flawless obtaining 100% true positive results while attaining 0% false positive detections. In another more comprehensive experiment the results show the least amount of false positive detections (28%) when compared with state-of-the-art methods, while obtaining 71% true positive results, and having overall the best compromise between true positive and false positive as the DIS metric (0.40) shows. Observing the achieved results, we believe that such a system has a good potential application in the surveillance of cluttered environments, such as industrial plants, where their economic operation can greatly benefit from automated operations.

Yet more work can be done to improve the already good results of the proposed technique. Among the ideas that will follow up the developments of the present work are: the elaboration of a technique to perform more generic video alignment that does not rely on a specific type of movement by using dynamic time warping (DTW) techniques on visually extracted features (da Silva et al. 2017); also to allow the video registration to be applied on such broad scenarios the homography technique should be replaced with other more comprehensive methods such as registration via 3D features obtained via the fundamental matrix computation (Hartley and Zisserman 2003). Other ideas that could be applied to the current method are the illumination compensation via color space transformations to allow the algorithm to cope with illumination changes and other implementation optimizations to increase the number of frames that can be computed in real time.

Acknowledgements This work was developed with the partial support of Statoil Brazil, Petrobras, ANP, and CAPES and CNPq funding agencies.

Funding

Funding was provided by Conselho Nacional de Desenvolvimento Científico e Tecnológico (Grant No. 203876/2017-2), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Grant No. 88881.135449/2016-01), Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro.

References

Chang, L., Zhao, H., Zhai, S., Ma, Y., & Liu, H. (2013). Robust abandoned object detection and analysis based on online learning. In *International conference on robotics and biomimetics, Shenzhen, China*.

Chen, Y. M., & Bajie, I. V. (2011). A joint approach to global motion estimation and motion segmentation from a coarsely sampled motion vector field. *IEEE Transactions on Circuits and Systems for Video Technology*, *21*(9), 1316–1328.

Cheng, L., Gong, M., Schuurmans, D., & Caelli, T. (2011). Real-time discriminative background subtraction. *IEEE Transactions on Image Processing*, *20*(5), 1401–1414.

Choi, W., Pantofaru, C., & Savarese, S. (2013). A general framework for tracking multiple people from a moving camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(7), 1577–1591.

da Silva, A. F., Thomaz, L. A., Carvalho, G., Nakahata, M. T., Jardim, E., de Oliveira, J. F. L., et al. (2014). An annotated video database for abandoned-object detection in a cluttered environment. In *International telecommunications symposium, Sao Paulo, Brazil*.

da Silva, A. F., Thomaz, L. A., Netto, S. L., & da Silva, E. A. B. (2017). Online video-based sequence synchronization for moving camera object detection. In *IEEE international workshop on multimedia signal processing* (pp. 1–6). Luton, UK.

Davis, J. W., Morison, A. M., & Woods, D. D. (2007). An adaptive focus-of-attention model for video surveillance and monitoring. *Machine Vision and Applications*, *18*(1), 41–64.

DeSouza, G. N., & Kak, A. C. (2002). Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(2), 237–267.

Dore, A., Soto, M., & Regazzoni, C. S. (2010). Bayesian tracking for video analytics. *IEEE Signal Processing Magazine*, *27*(5), 46–55.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(9), 1627–1645.

FLIR. (2016). Flir GF320. http://www.flir.com/ogi/display/?id=55671. Accessed January 14, 2016.

Ghosh, A., NSubudhi, B., & Ghosh, S. (2012). Object detection from videos captured by moving camera by fuzzy edge incorporated Markov random field and local histogram matching. *IEEE Transactions on Circuits and Systems for Video Technology*, *22*(8), 1127–1135.

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision* (2nd ed.). Cambridge: Cambridge University Press.

Hu, W. C., Chen, C. H., Chen, T. Y., Huang, D. Y., & Wu, Z. C. (2015). Moving object detection and tracking from video captured by moving camera. *Journal of Visual Communication and Image Representation*, *30*, 164–180.

iRobot. (2016). iRobot Roomba vacuum cleaning robot. http://www.irobot.com/For-the-Home/Vacuuming/Roomba.aspx. Accessed February 26, 2018.

Jodoin, P. M., Saligrama, V., & Konrad, J. (2012). Behavior subtraction. *IEEE Transactions on Image Processing*, *21*(9), 4244–4255.

Kim, J., Ye, G., & Kim, D. (2010). Moving object detection under free-moving camera. In *IEEE international conference on image processing, Hong Kong*.

Kim, C., & Hwang, J. N. (2002). Fast and automatic video object segmentation and tracking for content-based applications. *IEEE Transactions on Circuits and Systems for Video Technology*, *12*(2), 122–129.

Kim, S., Yun, K., Yi, K., Kim, S., & Choi, J. (2013). Detection of moving objects with a moving camera using non-panoramic background model. *Machine Vision and Applications*, *24*(5), 1015–1028.

Kong, H., Audibert, J. Y., & Ponce, J. (2010). Detecting abandoned objects with a moving camera. *IEEE Transactions on Image Processing*, *19*(8), 2201–2210.

Kucharczak, F., da Silva, A. F., Thomaz, L. A., Carvalho, G., da Silva, E. A. B., & Netto, S. L. (2014). Comparison and optimization of image descriptors for real-time detection of abandoned objects. In *Simpósio de Processamento de Sinais da UNICAMP, Campinas, Brazil*. http://www.sps.fee.unicamp.br/anais/vol01/VSPS_a24_LThomaz.pdf.

Kundu, A., Jawahar, C. V., & Krishna, K. M. (2010). Realtime moving object detection from a freely moving monocular camera. In *IEEE international conference on robotics and biomimetics* (pp. 1635–1640). Tianjin, China.

Lee, S., Yun, I. D., & Lee, S. U. (2010). Robust bilayer video segmentation by adaptive propagation of global shape and local appearance. *Journal of Visual Communication and Image Representation*, *21*(7), 665–676.

Lin, Y., Tong, Y., Cao, Y., Zhou, Y., & Wang, S. (2017). Visual-attention based background modeling for detecting infrequently moving objects. *IEEE Transactions on Circuits and Systems for Video Technology*, *27*(6), 1208–1221.

Li, H., Tang, J., Wu, S., Zhang, Y., & Lin, S. (2010). Automatic detection and analysis of player action in moving background sports video sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, *20*(3), 351–364.

Menezes, P., Lerasle, F., & Dias, J. (2011). Towards human motion capture from a camera mounted on a mobile robot. *Image and Vision Computing*, *29*(6), 382–393.

Micheloni, C., & Foresti, G. L. (2006). Real-time image processing for active monitoring of wide areas. *Journal of Visual Communication and Image Representation*, *17*(3), 589–604.

Mukojima, H., Deguchi, D., Kawanishi, Y., & Ide, I. (2016). Moving camera background subtraction for obstacle detection on railway tracks. In *International conference on image processing, Arizona, USA*.

Nakahata, M. T., Thomaz, L. A., da Silva, A. F., da Silva, E. A. B., & Netto, S. L. (2017). Anomaly detection with a moving camera using spatio-temporal codebooks. *Multidimensional Systems and Signal Processing*. https://doi.org/10.1007/s11045-017-0486-8.

Nawaz, T., Poiesi, F., & Cavallaro, A. (2014). Measures of effective video tracking. *IEEE Transactions on Image Processing*, *23*(1), 376–388.

Nordlund, P., & Uhlin, T. (1996). Closing the loop: Detection and pursuit of a moving object by a moving observer. *Image and Vision Computing*, *14*(4), 265–275.

Pinto, A. M., Correia, M. V., Moreira, A. P., & Costa, P. G. (2014). Unsupervised flow-based motion analysis for an autonomous moving system. *Image and Vision Computing*, *32*(6–7), 391–404.

Romanoni, A., Matteucci, M., & Sorrenti, D. G. (2014). Background subtraction by combining temporal and spatio-temporal histograms in the presence of camera movement. *Machine Vision and Applications*, *25*(6), 1573–1584.

Saligrama, V., Konrad, J., & Jodoin, P. M. (2010). Video anomaly identification. *IEEE Signal Processing Magazine*, *27*(5), 18–33.

Soille, P. (2003). *Morphological image analysis: Principles and applications* (2nd ed.). Berlin: Springer.

Subudhi, B. N., Nanda, P. K., & Ghosh, A. (2011). A change information based fast algorithm for video object detection and tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, *21*(7), 993–1004.

Suhr, J. K., Jung, H. G., Li, G., Noh, S. I., & Kim, J. (2011). Background compensation for pan-tilt-zoom cameras using 1-D feature matching and outlier rejection. *IEEE Transactions on Circuits and Systems for Video Technology*, *21*(3), 371–377.

Sun, S. W., Wang, Y. C. F., Huang, F., & Liao, H. Y. M. (2013). Moving foreground object detection via robust SIFT trajectories. *Journal of Visual Communication and Image Representation*, *24*(3), 232–243.

Taneja, A., Ballan, L., & Pollefeys, M. (2015). Geometric change detection in urban environments using images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(11), 2193–2206.

Tian, Y., Feris, R., Liu, H., Humpapur, A., & Sun, M. T. (2011). Robust detection of abandoned and removed objects in complex surveillance videos. *IEEE Transactions on Systems, Man, and Cybernetics*, *41*(5), 565–576.

Tomioka, Y., Takara, A., & Kitazawa, H. (2012). Generation of an optimum patrol course for mobile surveillance camera. *IEEE Transactions on Circuits and Systems for Video Technology*, *22*(2), 216–224.

VDAO. (2014). VDAO—Video database of abandoned objects in a cluttered industrial environment. http://www.smt.ufrj.br/~tvdigital/database/objects. Accessed February 26, 2018.

VDAO-200. (2017). 200-frame excerpts form VDAO database. http://www02.smt.ufrj.br/~tvdigital/database/research/. Accessed February 26, 2018.

Weinzaepfel, P., Revaud, J., Harchaoui, Z., & Schmid, C. (2013). Deppflow: Large displacement optical flow with deep matching. In *International conference on computer vision, Sydney, Australia*.

Xie, Y., Lin, L., & Jia, Y. (2010). Tracking objects with adaptive feature patches for PTZ camera visual surveillance. In *International conference on pattern recognition* (pp. 1739–1742). Istanbul, Turkey.

Xie, C., Tan, J., Chen, P., Zhang, J., & He, L. (2014). Multi-scale patch-based sparse appearance model for robust object tracking. *Machine Vision and Applications*, *25*(7), 1859–1876.

Xue, K., Ogunmakin, G., Liu, Y., Vela, P. A., & Wang, Y. (2011). PTZ camera-based adaptive panoramic and multi-layered background model. In *IEEE international conference on image processing, Brussels, Belgium*.

Xue, K., Liu, Y., Ogunmakin, G., Chen, J., & Zhang, J. (2013). Panoramic Gaussian mixture model and large-scale range background substraction method for PTZ camera-based surveillance systems. *Machine Vision and Applications*, *24*(3), 477–492.

Yilmaz, A. (2011). Kernel-based object tracking using asymmetric kernels with adaptive scale and orientation selection. *Machine Vision and Applications*, *22*(2), 255–268.

Gustavo H. F. de Carvalho was born in Rio de Janeiro, Brazil. He received the B.Sc. degree (cum laude) in Electronic and Computer Engineering from the Universidade Federal do Rio de Janeiro (UFRJ), Brazil, in 2005, the M.Sc. degree in Systems and Computer Engineering from COPPE/UFRJ in 2009, and the D.Sc. degree in Electrical Engineering from COPPE/UFRJ in 2015. His research interests include the areas of video and image processing, and computer vision.

Lucas A. Thomaz was born in Niterói, Brazil. He received the B.Sc. (cum laude) degree in electronic and computer engineering from the Universidade Federal do Rio de Janeiro (UFRJ), Brazil, in 2013, and the M.Sc. degree in electrical engineering from COPPE/UFRJ in 2015, where he is currently pursuing the Ph.D. degree at the Program of Electrical Engineering. Since 2017, he has been a Visiting Researcher Scholar with North Carolina State University. His research interests include the areas of computer vision, digital signal processing, video, and image processing.

Allan F. da Silva was born in Brazil in 1990. He received the B.Sc. degree in electronic and computer engineering and the M.Sc. degree in electricalengineering from the Universidade Federal do Rio de Janeiro in 2013 and 2015, where he is currently pursuing the Ph.D. degree. Since 2017, he has been a Visiting Researcher with the Université de Bordeaux. His research interests include the areas of computer vision, video, and image processing.

Eduardo A. B. da Silva was born in Rio de Janeiro, Brazil. He received the Electronics Engineering degree from the Instituto Militar de Engenharia, Rio de Janeiro, in 1984, the M.Sc. degree in electrical engineering from the Universidade Federal do Rio de Janeiro in 1990, and the Ph.D. degree in electronics from the University of Essex, UK, in 1995. He was with the Department of Electrical Engineering, Instituto Militar de Engenharia, in 1987 and 1988. He has been with the Department of Electronics Engineering, UFRJ, since 1989, and with the Department of Electrical Engineering, COPPE/UFRJ, since 1996. He is the Co-Author of the book Digital Signal Processing—System Analysis and Design (Cambridge University Press, 2002) that has also been translated to the Portuguese and Chinese languages, whose second edition has been published in 2010. His research interests lie in the fields of signal and image processing, signal compression, and digital TV and pattern recognition, together with its applications to telecommunications and the oil and gas industry. He was Technical Program Co-Chair of ISCAS2011. He has served as an Associate Editor of the IEEE Transactions on Circuits and Systems I and II and Multidimensional, Systems and Signal Processing. He was the Deputy Editor-in-Chief of the IEEE Transactions on Circuits and Systems I in 2016 and 2017 and is Associate Editor of the Journal of the Franklin Institute. He has been a Distinguished Lecturer of the IEEE Circuits and Systems Society.

Sergio L. Netto was born in Rio de Janeiro, Brazil. He received the B.Sc. (cum laude) degree from the Universidade Federal do Rio de Janeiro (UFRJ), Brazil, in 1991, the M.Sc. degree from COPPE/UFRJ in 1992, and the Ph.D. degree from the University of Victoria, BC, Canada, in 1996, all in electrical engineering. Since 1997, he has been with the Department of Electronics and Computer Engineering, Poli/UFRJ, and since 1998, he has been with the Program of Electrical Engineering, COPPE/UFRJ. He is the Co-Author (with P. S. R. Diniz and E. A. B. da Silva) of Digital Signal Processing: System Analysis and Design (Cambridge University Press, second edition, 2010). His research and teaching interests lie in the areas of digital signal processing, speech processing, information theory, and computer vision.