

Anomaly detection with a moving camera using spatio-temporal codebooks

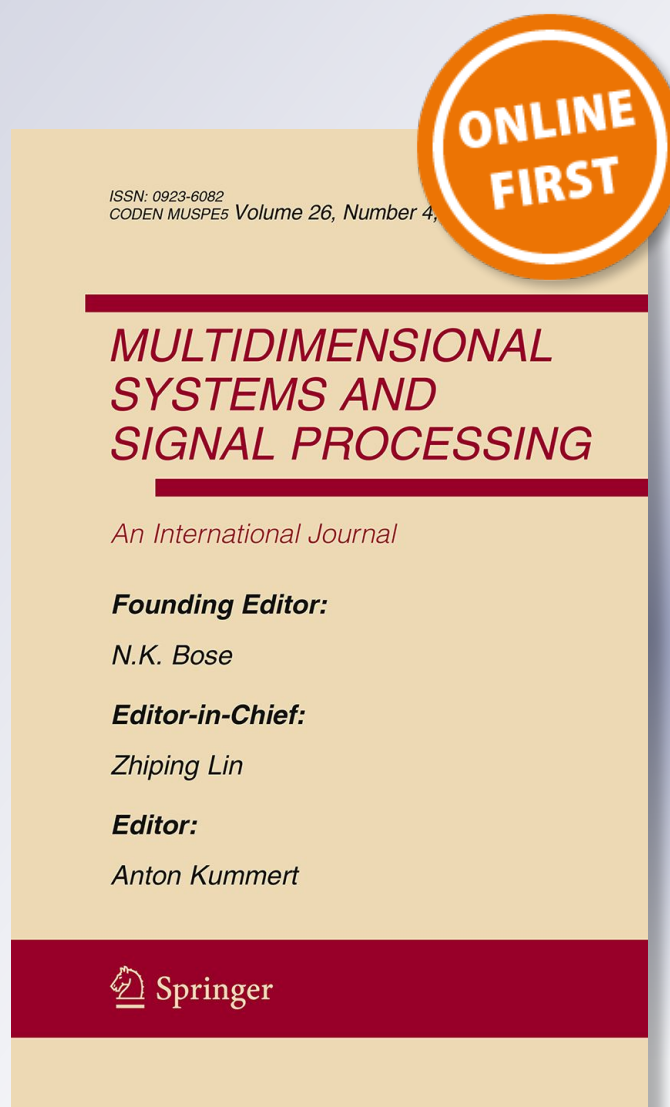
**Mateus T. Nakahata, Lucas A. Thomaz,
Allan F. da Silva, Eduardo A. B. da Silva
& Sergio L. Netto**

**Multidimensional Systems and Signal
Processing**

An International Journal

ISSN 0923-6082

Multidim Syst Sign Process
DOI 10.1007/s11045-017-0486-8



Your article is protected by copyright and all rights are held exclusively by Springer Science +Business Media New York. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com".

Anomaly detection with a moving camera using spatio-temporal codebooks

Mateus T. Nakahata¹ · Lucas A. Thomaz¹ · Allan F. da Silva¹ ·
Eduardo A. B. da Silva² · Sergio L. Netto²

Received: 16 August 2016 / Revised: 19 February 2017 / Accepted: 16 March 2017
© Springer Science+Business Media New York 2017

Abstract This paper proposes a method to detect anomalies in videos acquired by a camera mounted on a moving inspection robot. The proposed method is based on a spatio-temporal composition (STC) method, where a dense sampling is used to break the video into small 3D volumes that are used to calculate the probability of the spatio-temporal arrangements. This class of methods has been successfully used for surveillance videos obtained by static cameras. However, when applied to videos recorded by cameras on moving platforms, the STC gives a large number of false detections. In this work, we propose improvements to the present STC method that will alleviate this problem in two ways. First, a two-stage dictionary learning process is performed to allow a more reliable anomaly detection. Second, improved spatio-temporal features are employed. These modified features are extracted after an enhanced temporal filtering that performs a temporal regularization of the video sequence. The proposed approach gives very good results in the identification of anomalies without the need of background subtraction, motion estimation or tracking. The results are shown to be comparable or even superior to those of other state-of-the-art methods using bag-of-video words or other moving-camera surveillance methods. The system is accurate even with no prior knowledge of the type of event to be observed, being robust to cluttered environments, as illustrated by several practical examples. These results are obtained without compromising the performance of the algorithm in the static cameras case.

Keywords Video surveillance · Moving camera · Abandoned object detection · Cluttered environment · Spatiotemporal composition · Bag of video words

✉ Mateus T. Nakahata
mateus.nakahata@smt.ufrj.br

¹ PEE/COPPE, Universidade Federal do Rio de Janeiro, Av. Horácio Macedo, 2030, Centro de Tecnologia, bloco H, sala 221, Cidade Universitária, Rio de Janeiro, RJ CEP 21941-598, Brazil

² DEL/POLI-PEE/COPPE, Universidade Federal do Rio de Janeiro, Av. Horácio Macedo, 2030, Centro de Tecnologia, bloco H, sala 221, Cidade Universitária, Rio de Janeiro, RJ CEP 21941-598, Brazil

1 Introduction

Video surveillance systems have been widely used in public and private security (Haering et al. 2008). They allow a remote inspection of the environment, while removing the necessity of direct exposure to dangerous environments and reducing the need of human on-site presence. On the other hand, the surveillance-system operators are exposed to a huge amount of images on a 24/7 basis. In such conditions, it is difficult to keep the focus for more than a few minutes, and after short periods of time the efficiency of human-operated video surveillance systems decreases by a significant amount. In order to minimize this problem, automated systems for video analysis are among the preferred solutions. However, these systems need to be trained and configured to work properly. This requires prior knowledge of the events of interest, which is not often possible. Furthermore, often the monitored environments are cluttered or changing over time, and the video analysis systems need to be constantly reconfigured. Most of the methods used for this purpose perform a tracking of objects and background subtraction. Some methods propose different representations to characterize the frames and their objects (e.g. image descriptors) and use these representations to extract important information about them. However, most of these methods do not have a good performance in cases where there is a great variability in the anomaly type or the scene is visually cluttered.

Among the most challenging surveillance applications are the detection of targets and objects in remote sensing images. These applications aim to detect the presence of objects and targets in images from the surface of the planet obtained with very high resolution cameras that are usually placed on satellites. A variety of methods address such applications, and several of those can be found in the survey presented in Cheng and Han (2016). Some employ deep neural-network techniques, which require a very precise setup of its search areas (regional ratios), partition block, input-frame and filtering-array characteristics, number of layers, and so on, to work properly.

Particularly, Zhou et al. (2016) searches for specific classes of objects included in a pre-determined training set. A different approach presented in Cheng et al. (2016) includes rotation-invariant techniques to add robustness to the object detection. Although excellent results have been obtained with such methods, both of them detect only objects specified during the systems training phase. Therefore, these two approaches are not suitable for the objectives of the present work, that is to detect anomalous events without any prior knowledge about them.

Boiman and Irani (2007) detect anomalous events by reconstructing the video from previous samples while maximizing the probability of every video chunk to compose the present sequence. Each video is decomposed into multiple spatio-temporal volumes and then reconstructed using only the volumes previously observed using dense sampling. The spatio-temporal arrangement of the neighboring volumes is also taken into consideration. However, this method has the drawback of being unable to execute in real time due to the large computational complexity associated with dense sampling and the corresponding large number of reference samples generated.

One approach that has been used in video analysis is the bag-of-video words (BOV) method (Liu and Shah 2008). In it, the data is represented as small spatio-temporal volumes, with the redundancy among them being minimized through the use of a codebook (Lazebnik et al. 2006). This codebook is then used to analyze the videos for detecting anomalies. Such methods tend to perform well in cluttered environments. However, the BOV approach does not consider the influence of the spatio-temporal composition of the objects that compose the videos, which is considered crucial for human image interpretations (Schwartz et al. 2007).

The spatio-temporal composition method (STC) (Roshtkhari and Levine 2013) considers a spatio-temporal array of small video volumes and models it by using a probabilistic approach. In it, anomalous events are those with a low probability of occurrence. Some interesting features of the STC method are that it can be trained on-line, having the characteristic of adapting as environmental conditions change. In addition, it requires little or even no pre-settings for the detection of anomalies and is fast enough to be used in real-time applications. Although it performs very well for anomaly detection in videos recorded with static cameras, the STC usually fails when the camera is moving, as in this case the background changes continuously and at a fast rate.

There are many other methods that are able to perform the detection of anomalies in video sequences, as detailed in the comprehensive survey presented in Cuevas et al. (2016). Although several methods are listed in this survey, very few of them are able to deal with videos acquired with moving cameras, which is still considered a very challenging problem. One example can be found in Suhr et al. (2011), where background subtraction is performed in videos obtained using a pan-tilt-zoom (PTZ) camera with the aid of a similarity matrix. The relationship between consecutive frames is approximated by a three-parameter similarity transformation, which is separable in the vertical and horizontal axes. The outliers are removed by the random sample consensus (RANSAC) (Fischler and Bolles 1981) method.

Zhou et al. (2009), detection of moving objects using a moving camera is performed by extracting the scale-invariant feature-transform (SIFT) (Lowe 2004) features and finding correspondences between those features in the frames. The RANSAC algorithm is then used to remove the outliers, background subtraction is applied to perform the object detection and a dynamic background modeling is used to improve the overall detection.

The work of Kong et al. (2010) describes the use of a camera mounted on a moving car to detect abandoned objects along a road. In that system, first, a reference video without abandoned objects is recorded along the path. Then, further passes of the camera along the same path are processed to find the presence of anomalous objects in the scene. The coordinates of a global positioning system (GPS) are used to align the reference and target videos. The salient points are obtained using SIFT, and a homography transformation is calculated using the RANSAC to estimate affine transformations between corresponding frames of the the video sequences. Then, the frames are compared using an experimentally obtained threshold on the normalized cross-correlation (NCC) image.

Another example of anomaly detection using a moving camera is the one in de Carvalho (2015). There, a camera is mounted on a robot that performs a translational movement. First, one records a reference video of what is considered a normal situation (without abandoned objects) along the path of the camera. Every new (target) video along the path is compared with this reference to detect abandoned objects. Salient points in the reference and observation images are obtained through the speeded-up robust features (SURF) algorithm (Bay et al. 2008). The videos are geometrically registered by means of a homography transformation using the detected salient points and the RANSAC algorithm. The NCC is applied between the two images (from the reference video and the transformed observation). After some post-processing a threshold is used to determine the areas in the frame associated to abandoned objects.

Moving cameras are also used in the work presented in Mukojima et al. (2016). It performs the detection of anomalies in train tracks using videos that are acquired by a camera mounted on the train that oversees the train's frontal path along the rail. To this end the method uses deep flow features (Weinzaepfel et al. 2013) to align the frames from two videos (a video containing a reference footage with no anomalies and a target video that may contain anomalies) and then computes two different similarity measurements to assess the presence

of anomalies in the scene. The final output is obtained by verifying the intersection between the results from the two distinct measurements.

Given the nice properties of the STC method, a natural development would be to apply it to the moving-camera case. Unfortunately, as will be shown later in this paper, the plain STC fails when applied to surveillance videos such as the ones used in [de Carvalho \(2015\)](#), obtained from a camera mounted on a moving robot. This motivated us to propose in this paper the STC-mc (STC-moving camera) method, which is a further development of the STC method that is able to adequately handling the moving camera case. Among the innovations introduced by the STC-mc method are the use of an enhanced spatio-temporal descriptor preceded by temporal smoothing of the videos, enabling the system to cope with variations in video parallax. The other main innovation brought by the STC-mc algorithm is the use of a two-stage dictionary learning process where a second codebook is used to reduce false detections. The assessment of the proposed STC-mc method is carried out using several videos from the publicly available VDAO database ([Silva et al. 2014](#)), which contains reference videos without abandoned objects as well as several versions of the same video with different abandoned objects.

To describe the proposed STC-mc algorithm, this paper is organized as follows: Sect. 2 presents a short description of the main steps of the STC algorithm and evaluates its performance in both the static- and moving-camera cases. Section 3 presents a description of the STC-mc algorithm, detailing its main steps and important parameters. Section 4 describes the VDAO database. Section 5 details the methodology used to tune the parameters of the proposed algorithm. In Sect. 6 the performance of the proposed algorithm is assessed using the VDAO database and its results are compared with the ones of other BOV and STC methods, as well as the ones of state-of-the-art video anomaly-detection methods. Finally, Sect. 7 concludes the paper emphasizing the main contributions involved in the development of the STC-mc algorithm.

1.1 Main contributions

The main contributions of the present work are modifications to the traditional STC method that allow it to cope with the challenge of detecting anomalies in videos acquired with moving cameras. The proposed modifications allow the proposed STC-mc algorithm to detect successfully static foreground objects in moving-camera videos, which is not a strong suit of the original STC algorithm. The three main contributions of the proposed STC-mc method are:

- Introduction of a Gaussian filtering to deal with misalignments caused by camera shaking (see Sect. 3.1);
- The use of a new descriptor based on both spatial and temporal gradients (see Sect. 3.2);
- Use of a second dictionary that is generated by performing the training in two stages (see Sect. 3.3).

Combined altogether these proposals allow the STC framework to be successfully employed in the detection of anomalies in moving-camera videos, as will be demonstrated later in the paper.

2 Spatio-temporal composition method

In this section, we make a brief description of the method proposed in [Roshtkhari and Levine \(2013\)](#) to find anomalous events in videos. We also provide details of our implementation

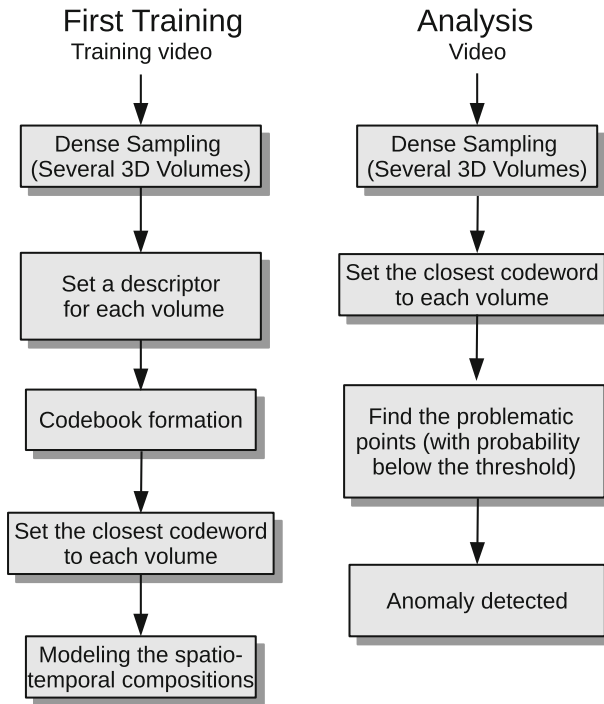


Fig. 1 The training and analysis steps of the STC method

of this method that will be used as reference for the development of our own method. In this method, new samples of video are broken down into small volumes that are represented by codewords from a codebook. Then, the probabilities of occurrence of spatio-temporal compositions formed by these codewords are calculated. Compositions with low probability are candidates to be anomalous. The training is conducted with a small sample video of a normal scene. The initial stages of sampling and creating the descriptors are identical in the training and analysis phases. Figure 1 shows the main steps of the STC method for identifying anomalies in images.

2.1 Dense sampling

The sampling is based on BOV, consisting of spatio-temporal volumes obtained by dense sampling of the videos. In the analysis of videos, dense sampling usually have a superior performance when compared with the simple random sampling, because it is able to maintain the relevant information of a video (Rapantzikos et al. 2009). In this method, dense sampling is carried out, dividing the original video into small 3D volumes, $v_i \in \mathbb{R}^{n_x \times n_y \times n_t}$, as shown in Fig. 2, where $n_x \times n_y$ is a small frame area and n_t is the length of a small time interval.

As proposed in Bertini et al. (2012), the method performs the sampling such that there is an overlap of at least 40% between adjacent volumes. This yields satisfactory results, achieving a compromise between accuracy and processing time. In our reference STC implementation, we adopted a spatial overlap of 50% + 1 pixel.

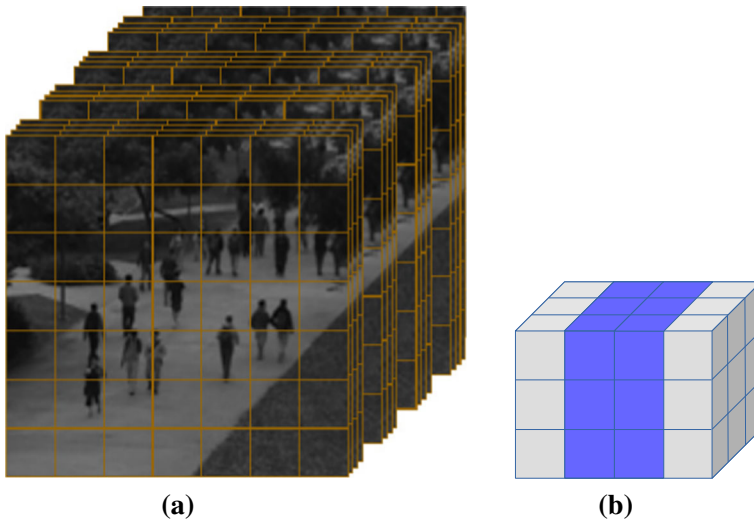


Fig. 2 **a** Dense sampling of a video. **b** Two $3 \times 3 \times 3$ pixel volumes with a 2-pixel overlap, or $50\% + 1$ (Color figure online)

2.2 3D volumes descriptor

Each volume v_i is represented by a descriptor g_i , which is defined as the absolute value of the time derivative Δ_t of each pixel in the volume v_i :

$$\forall v_i, g_i = \text{abs}(\Delta_t(v_i)). \tag{1}$$

The values obtained for each pixel of v_i are stacked in a vector. The volume v_i used in our implementation were of dimensions $7 \times 7 \times 5$ pixels, so the descriptor had a dimension of 1×245 , empirically defined following the rationale in Roshtkhari and Levine (2013). This descriptor is said to be robust even when dealing with videos from cluttered environment. Although this descriptor works well in many cases, other descriptors may perform better depending on the application, as given in Zhong et al. (2004) and Bertini et al. (2012).

2.3 Codebook

Due to the dense sampling, the number of spatio-temporal volumes becomes too large, but these volumes have a lot of redundancy among them. So, to decrease the complexity, similar volumes are clustered and for every group a codeword is created from these volumes descriptors. The codewords are saved in a codebook, which can be created using a clustering method, such as K -means (Heijden et al. 2004). In this work, however, the codebook creation is carried out using the algorithm described in Roshtkhari and Levine (2013), as detailed in Algorithm 1, where the only parameter to be set is the maximum distance ε_1 that separates the codewords from each other. In Sect. 5 the method utilized to find the best value of this distance is described.

After creating the codebook, each volume v_i is related to a codeword c_j with a weight $w_{i,j}$ given by

Algorithm 1 Codebook Creation algorithm.

The training video is broken down into small volumes v_i . α is a percentage of the distance. If two codes are at a distance below $\alpha\epsilon$, they can be merged.

Initialization

The first codeword is the first volume v_0 :

$$c_1 \leftarrow v_0$$

$$f_1 \leftarrow 1$$

$$P_{v_1} \leftarrow 1$$

Codebook Creation

The Euclidean distance is used to determine the similarity between the descriptor and the codewords in the codebook.

for All volumes $\{v_i\}_{i=1}^N$ **do**

if $\min_j d(v_i, c_j) > \epsilon$ **then**

 Create a new code: $c_{j+1} \leftarrow v_i$

else

 Calculate $w_{i,j}$ using: $w_{i,j} = \frac{1}{\sum_j \frac{1}{d(v_i, c_j)}} \times \frac{1}{d(v_i, c_j)}$

 Update the codebook: $c_j \leftarrow \frac{f_j \times c_j + w_{i,j} \times v_i}{f_j + w_{i,j}}$

 Update the frequency: $f_j \leftarrow f_j + 1$

 Calculate the prior probability: $P(c_j) = \frac{f_j}{N}$

end if

Pruning the Codebook

for All codewords $\{c_m\}_{m=1}^M$ **do**

if $\{d(c_i, c_j) < \alpha\epsilon, (0 < \alpha < 1)\}$ and $\{f_j < 0.1 \times \frac{N}{M}\}$ **then**

Merge the two codewords:

 Remove the codewords c_i e c_j from the codebook.

 Create a new codeword: $c_{M+1} \leftarrow \frac{f_i \times c_i + f_j \times c_j}{f_i + f_j}$

 Define the frequency of the new codeword: $f_{M+1} \leftarrow f_i + f_j$

end if

end for

end for

$$w_{i,j} = \frac{1}{\sum_j \frac{1}{d(v_i, c_j)}} \times \frac{1}{d(v_i, c_j)}, \tag{2}$$

where $d(v_i, c_j)$ is the Euclidean distance between the volume v_i and the codeword c_j .

2.4 Spatio-temporal composition

Most methods using BOV do not take into account the spatio-temporal arrangement between the volumes or limit it to a small volume around the sampling point. In the STC framework, a probabilistic approach is used to determine whether the volume is anomalous or not, based on the probability of the arrangement of the volumes within a larger region.

The representation of the set is made as follows: let E_i be the ensemble centralized at the point (x_i, y_i, t_i) in absolute coordinates and containing K volumes. This central point is used to determine the relative coordinates of the position of the volumes within the ensemble, according to Fig. 3a. Given the volume v_k in the set E_i , $\Delta_{v_k}^{E_i} \in \mathbb{R}^3$ is the relative position (in space and time) of v_k located at the point (x_k, y_k, t_k) within E_i :

$$\Delta_{v_k}^{E_i} = (x_k - x_i, y_k - y_i, t_k - t_i). \tag{3}$$

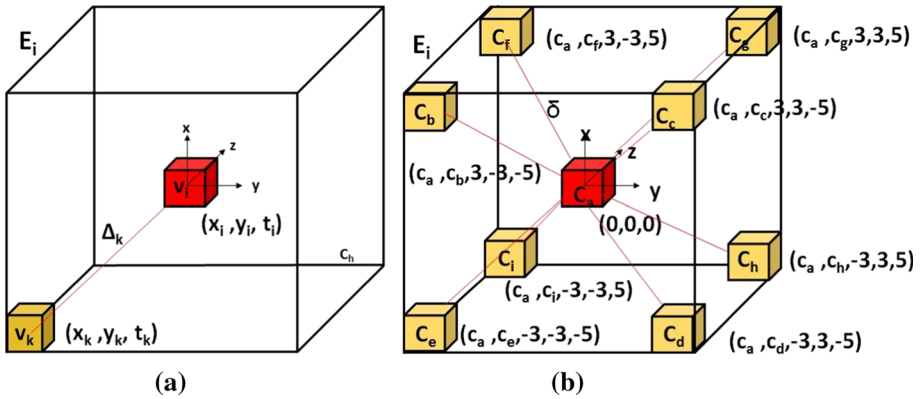


Fig. 3 **a** Relative position of the volumes in the set. The central volume v_i is at a distance Δ_k of the volume v_k . **b** After the substitution of the volumes by the closest code, the set is represented by spatio-temporal arrangement of codewords, which are at a distance δ of the central codeword c' . Example of a set at distance $\delta = \sqrt{43}$ contained in an ensemble of $7 \times 7 \times 10$ volumes (Color figure online)

Thus, the ensemble of volumes E_i , centered at position (x_i, y_i, t_i) , is initially represented as a set of video volumes and their relative positions with respect to the central volume:

$$E_i = \{\Delta_{v_k}^{E_i}, v_k, v_i\}_{k=1}^K. \tag{4}$$

Each volume v_k of the set is linked with the codeword $c_j \in \mathbf{C}$ with a weight w_j representing their similarity. Thus, the arrangement of volumes may be represented by a set of codewords and their spatio-temporal arrangement. Let $v \subset \mathbb{R}^{n_x \times n_y \times n_t}$ be the spatio of descriptors of a video volume, and \mathbf{C} the codebook; $c : v \rightarrow \mathbf{C}$ defines a random variable that allocates a codeword to a volume of video and $c' : v \rightarrow \mathbf{C}$ defines a random variable designating a codeword to the volume in the center of the ensemble. In addition, $\delta : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ defines a random variable representing the relative distance from the central point associated with codeword c' to the point associated with codeword c . Therefore, the ensemble of volumes can be represented as an arrangement of words of the codebook, as shown in Fig. 3b. In other words, instead of representing the E_i as an arrangement of volumes, it is represented as a codeword arrangement.

In this context, $O = (v_k, v_i, \Delta_{v_k}^{E_i})$ represents the observation of the volume v_k from the central volume v_i in the ensemble E_i , and $\Delta_{v_k}^{E_i}$ the relative position of the observed volume v_k with respect to v_i within E_i . The goal is to measure the probability $P(h|O)$ of each hypothesis $h = (c, c', \delta)$ obtained by replacing the volumes by codewords from the codebook, given the observation O , that is

$$P(h|O) = P(c, c', \delta | v_k, v_i, \Delta_{v_k}^{E_i}). \tag{5}$$

Roshtkhari and Levine (2013), it is shown that

$$P(c, c', \delta | v_k, v_i, \Delta_{v_k}^{E_i}) = P(\delta | v_k, v_i, \Delta_{v_k}^{E_i}) P(c' | v_i) P(c | v_k). \tag{6}$$

Hence, in an ensemble around a pixel, with a central volume v_i , and other volumes v_k within this ensemble at a distance $\Delta_{v_k}^{E_i}$ of the central volume, the aim is to calculate the probability of assigning the codeword c' to the central volume and c to other volumes. The probability $P(\delta | v_k, v_i, \Delta_{v_k}^{E_i})$ is determined by the approximation of its pdf by a mixture of Gaussians, using the expectation maximization algorithm (EM) (Bilmes 1998), where the samples are the codeword arrangements. In other words, the sample vector is of the form

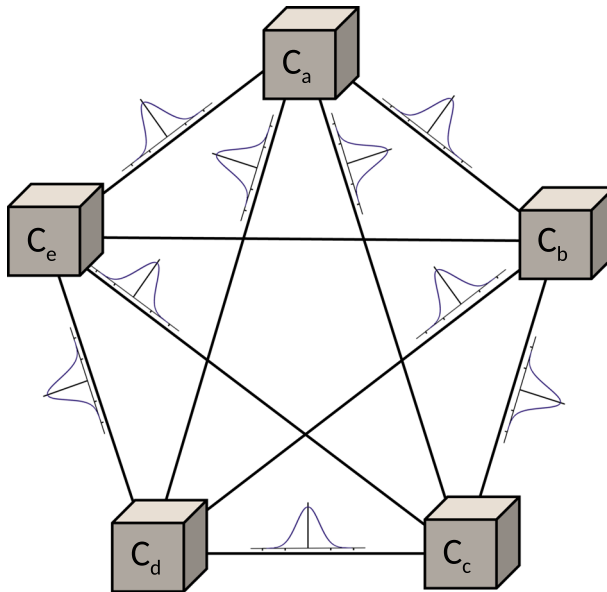


Fig. 4 The pdf of the 3D composition of the volumes v_k is calculated with the associated codewords c_k inside the ensemble E_i . The codebook is effectively formed by the codewords and the pdf

$\mathbf{a}(c_i, c_k, \delta)$, where δ is the relative distance between codewords. Several of these samples allow us to estimate the pdf, as shown in Fig. 4. The probabilities $P(c' | v_i)$ e $P(c | v_k)$ of each spatio-temporal volume are calculated during the allocation of codewords.

Therefore, in this method the dictionary is effectively formed not only by words but also by the distribution of the probabilities of the arrangements around each volume in the ensemble.

To calculate the parameters of the mixture of Gaussian using the EM algorithm, the number of Gaussians used in our implementation was three. The samples used to find the parameters formed a vector of dimension 1×5 , composed as follows: let E_i be the ensemble with a central volume v_i at the position (x_i, y_i, t_i) in absolute coordinates and containing K volumes. The relative coordinates (x_k, y_k, t_k) of the neighboring volumes v_k inside the ensemble are calculated from this point; v_i is represented by a codeword c_i and v_k by a codeword c_k ; j_{c_i} is the index of c_i in the first codebook and j_{c_k} is the index of c_k . Using these definitions, the first element is the index j_{c_k} of v_k , the second is the index j_{c_i} of v_i and the last three elements of the sample are the relative coordinates (x_k, y_k, t_k) . Therefore, the pdf is calculated using the relative position inside the ensemble and the codewords, represented by their index in the codebook.

2.5 Anomalous pattern detection

In the analysis phase the steps of sampling and descriptor calculation are the same as in the training phase. Then, using the codebook created in the training phase, the distance between the volume and every codeword is computed using Eq. (2).

Equation (6) is the codeword probability assignment, that is dependent on the relations between the central volume and the other volumes v_k in the ensemble E_i . Given a video of interest V , E_i^V is an ensemble of video volumes centered at point (x_i, y_i, t_i) and v_i is the central volume of this ensemble. The probability of the volume v_i can be written as:

$$P(c, c', \delta | E_i^V) = \prod_k^K P\left(\delta | c, c', \Delta_{v_k}^{E_i^V}\right) P(c | v_k) P(c' | v_i), \tag{7}$$

where v_k is a volume inside E_i^V , $\Delta_{v_k}^{E_i^V}$ is the relative position of the volume v_k , c' is the codeword attributed to v_i , c is the codeword attributed to v_k and δ is the relative distance of the codeword in the codeword space. The term $P(\delta | c, c', \Delta_{v_k}^{E_i^V})$ is the probability of the spatio-temporal arrangement, whose pdf is calculated as given in Sect. 2.4.

The a posteriori probability is calculated according to

$$P(c_j | v_i) = \frac{w_{i,j} \times P(c_j)}{\sum_j w_{i,j} \times P(c_j)}, \tag{8}$$

where the weight w is given by Eq. (2).

In brief, the video V to be analyzed is densely sampled into video volumes v_i . For each v_i a codeword $c_k \in C$ is allocated with a similarity $w_{i,j}$. The probability $P(c, c', \delta | E_i^V)$ of each volume to be an anomaly is calculated based on the spatio-temporal arrangement of the volumes within the ensemble E_i^V , centered in v_i . For each volume the probability of occurrence is computed using Eq. (7). Volumes with a probability smaller than a given threshold, obtained experimentally, are considered to be anomalous. Figure 5 illustrates the use of the threshold. Ideally, only the anomalous points have a probability below the threshold.

2.6 Results and conclusions

To illustrate the performance of the STC method, simulations were performed using the database provided in UCSD (2014). Initially, the training was made with a short video of about ten seconds where there were just people walking. In the test video there were people walking too, but there was also an anomaly consisting of a person riding a bicycle. The results obtained for the first test video are shown in Fig. 6, where one can notice that STC performed quite well, as only the cyclist was detected.

The STC method has also been applied to the video database of abandoned objects (VDAO) (Silva et al. 2014) (see Sect. 4). The training phase has been performed using a reference video of the environment recorded from a camera mounted on a moving robot that performs a back and forth rectilinear movement. The test was performed using a video recorded in similar conditions, but with abandoned objects added to the environment. Figure 7 illustrates the STC performance when detecting a dark blue box and a shoe over a cluttered background. One can note that STC fails in these cases, generating many false detections. We have observed in our experiments that if the detection threshold is modified in order to avoid the false detections, the STC is not able to detect the abandoned objects.

The results in Fig. 7 suggest that the STC method is able to detect anomalies when there are movements in the scene, and the camera is static. However, it fails in the VDAO database, that concerns detection of abandoned objects over a cluttered background using a moving camera. One of the main purposes of this paper is to investigate a solution to this problem. This is done in the next section, where we propose the novel algorithm STC-mc (STC-moving camera).

3 The STC-moving camera algorithm: proposed methodology

In this section, we propose the STC-moving camera (STC-mc) algorithm. It is a new anomaly detection algorithm based on the same principles of the STC but with enhancements that allow

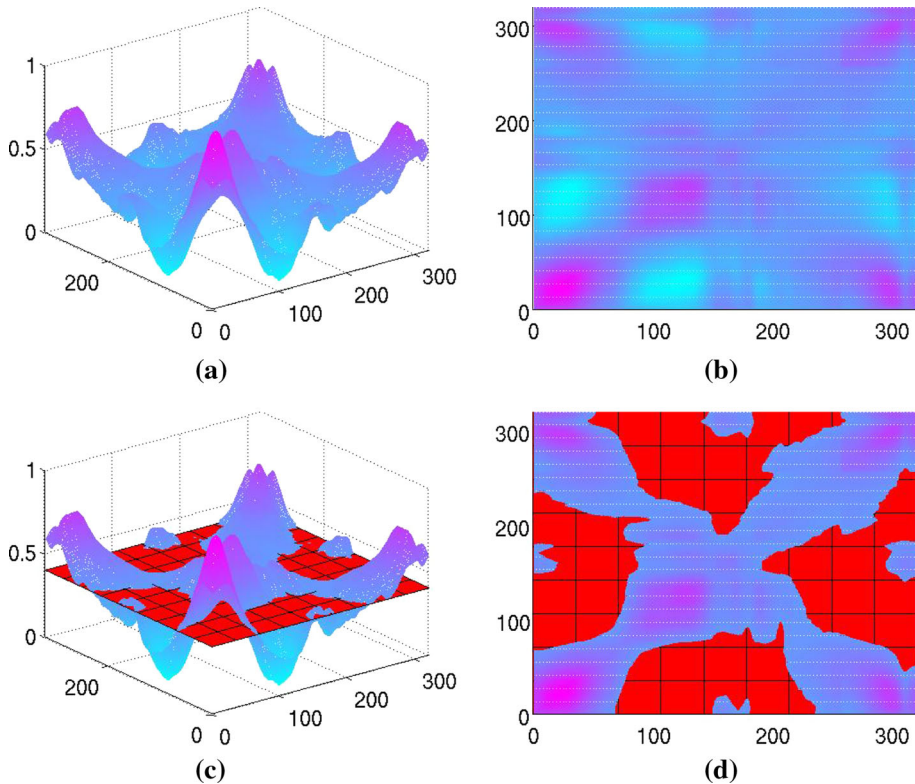


Fig. 5 Example of the use of the probability threshold. In the figures, the function $P(i, j)$ is the probability of the image points $I(i, j)$. Blue values represent points of low probability and purple ones represent points of high probability. Figures a, b represent a map of the probability of a sample frame. These maps have been cut by a plane representing the identification threshold. The higher the threshold, the more points are identified as being possibly anomalous. The points that are candidates to be anomalous are represented in red. a Target image. b Upper view. c Target image. d Upper view (Color figure online)

it to perform well in the detection of abandoned objects using videos acquired from a camera mounted on a moving platform. Figure 8 shows the main steps of STC-mc, highlighting its main contributions relative to the original STC. The steps corresponding to blocks in gray are the same as the ones in the STC method and the steps surrounded by a dotted rectangle represent the STC steps that have been enhanced in the STC-mc. The steps corresponding to the white blocks are not present in the original STC, being entirely proposed in this work. In what follows these blocks are described in detail.

3.1 Gaussian filtering

As seen in Eq. (1), the STC descriptor of a spatio-temporal volume is given by the time derivative of each pixel in the volume. When the camera is moving, it tends to shake along the path, specially if it is mounted on a platform that moves over an irregular rail. This may cause a large random variation on the values of this derivative from frame to frame, as illustrated in Fig. 9a, b, making this descriptor unsuitable for the task at hand.

To attenuate such variation in the temporal derivative, we propose to perform a temporal smoothing prior to the derivative computation. This can be done by employing a Gaussian

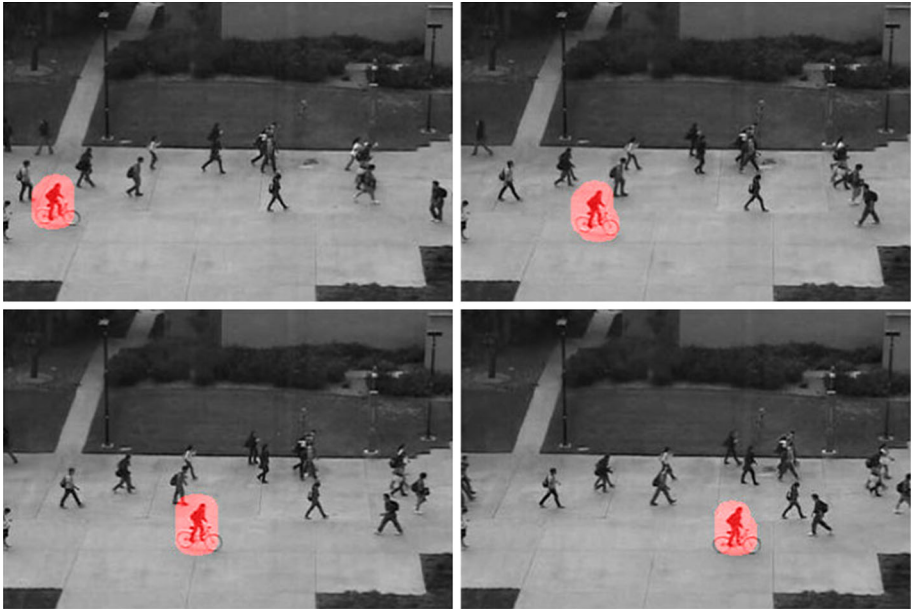


Fig. 6 Example of STC result with static camera. In this successful case, only the cyclist was detected. The people walking were not detected because in the training video there was people walking in a similar way (Color figure online)

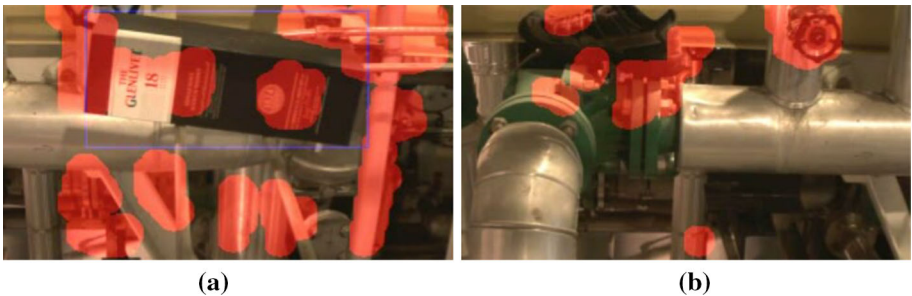


Fig. 7 Example of STC results obtained with a camera mounted on a moving robotic platform. In most of the frames, it was not possible to find a threshold where only the abandoned object was detected. **a** *Dark blue box*. **b** *Shoe* (Color figure online)

filter. The size of the filter kernel was set to 5 and the value of the standard deviation σ was tuned as described in Sect. 5.

Figure 9c, d show the time derivatives of the frames of Fig. 9a, b, respectively, after the temporal Gaussian filtering. Clearly this derivative is much more stable, and thus suitable for being incorporated in a spatio-temporal descriptor.

3.2 Enhanced spatio-temporal descriptor

As described in Sect. 2.2, the sampling of the video content is based on the BOV framework, which consists of spatio-temporal volumes obtained through dense sampling. The next step is to create a codebook to reduce redundancy between video volumes. For this reason, the

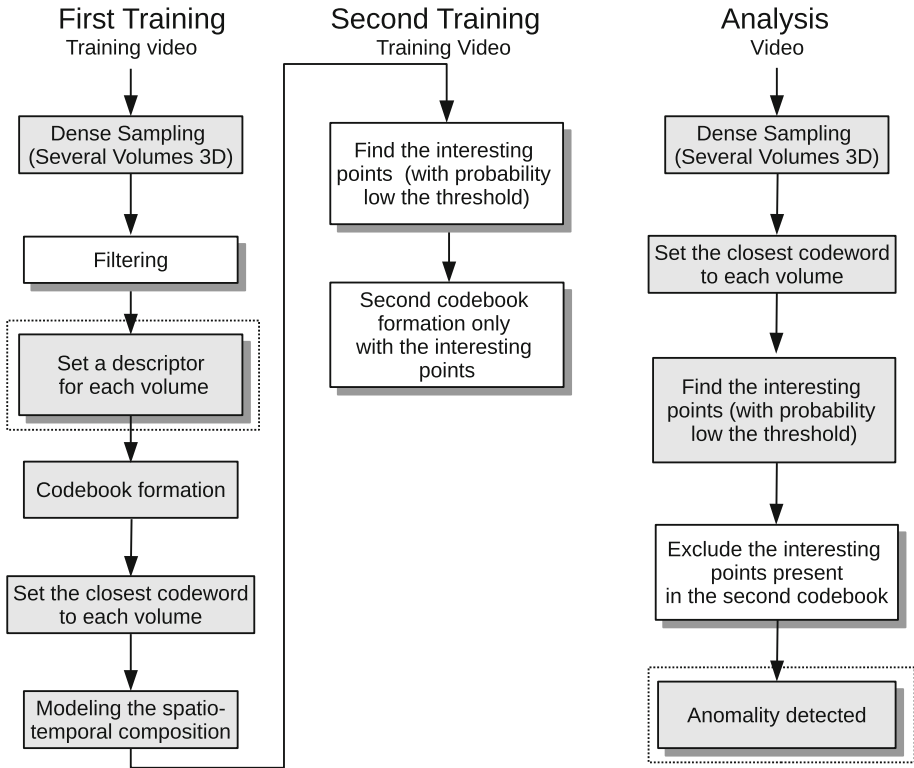


Fig. 8 The training and analysis steps of the STC method. The gray blocks are part of the original STC method. The gray blocks surrounded by a dotted rectangle represent the STC blocks where enhancements are proposed in this paper. The white blocks represent novel steps proposed in this paper

video is divided into small 3D volumes, where $n_x \times n_y$ is a small area and n_t is the length of a small time interval. Each volume v_i is represented by a descriptor. To choose a good descriptor for our application, a key issue to be taken into account is that, although the camera moves, the abandoned objects do not move relatively to the background. Thus, the relative motion of the objects in the scene was only caused by the parallax shift due to the movement of the camera. Since the parallax shift is a spatio-temporal effect, a descriptor based only in the temporal derivative as used by Boiman and Irani (2007) and Roshtkhari and Levine (2013) is not the most adequate. Therefore, we propose to use a descriptor which is based on a spatio-temporal derivative, as defined by the equation

$$D = \sqrt{\left(\frac{dI}{dx}\right)^2 + \left(\frac{dI}{dy}\right)^2 + \left(\frac{\lambda dI}{dt}\right)^2}. \tag{9}$$

To construct such descriptor, the value of D is computed for each pixel inside the volume v_i and the results are stacked in a vector. Note that this descriptor extracts useful information even in the more challenging case of environments where the background is cluttered and not static. In this new descriptor, it is important to observe that the temporal derivative is multiplied by a constant λ , whose value is yet to be determined as described in Sect. 5. This

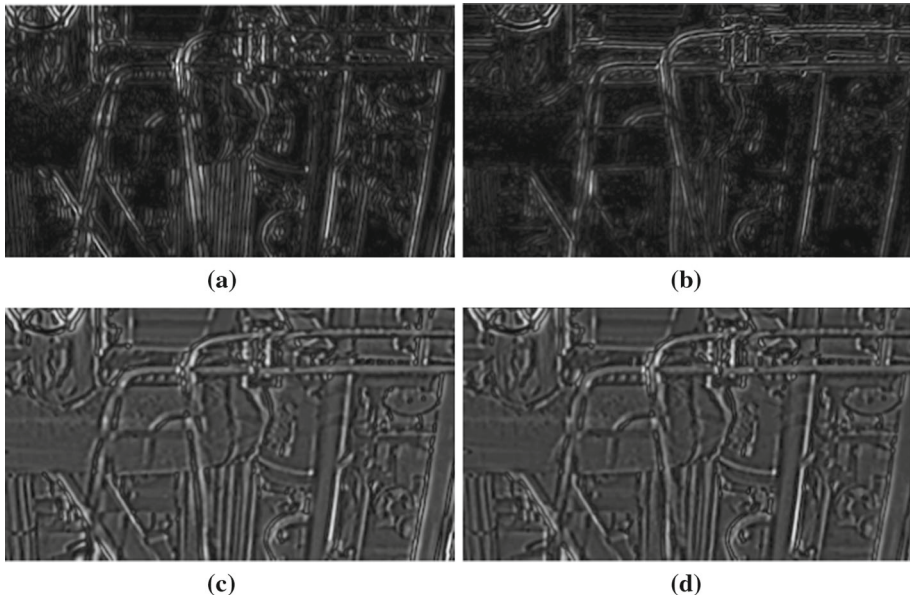


Fig. 9 **a, b** The temporal derivatives of two consecutive frames from the VDAO database as generated by the STC method [(a) Frame 1, (b) Frame 2]. One can see the large variation of this descriptor. **c, d** The temporal derivative after a Gaussian filtering [(c) Frame 1, (d) Frame 2]

constant performs an adjustment to account for relative effects of the spatial resolution, frame rate and Gaussian smoothing on the computation of the descriptor.

In our system, the volume size in pixels was empirically defined, according to [Roshtkhari and Levine \(2013\)](#). This led to a volume v_i equal to $n_x \times n_y \times n_t = 7 \times 7 \times 5$, such that the corresponding descriptor had a dimension of 1×245 .

3.3 Second codebook

When applying the original STC method to the videos in the VDAO database, often it is not possible to find a threshold which allows that only anomalous events have a probability value below it. That fact produces a large number of false detections. This happens because in the VDAO database there are cases when the probability of an anomaly (abandoned object) can be lower than the one of some points in the background.

In order to solve this problem we propose a two-stage training process for STC-mc. In it, a second codebook is introduced, containing the descriptors of the points where the probability is below a threshold. This additional codebook is used to exclude these points from the target video, so avoiding false detections. This is performed as follows: the main STC codebook is generated during the training phase, by processing the reference video, as in the original STC method. Then the reference video is processed again, this time to detect abandoned objects. Since, by definition, the reference video has no abandoned objects, ideally there should be no detections. However, this is not the case. There are several false detections along the reference video. Then, these volumes that have been wrongly detected, that is, the ones with their probability below the initial threshold, have their descriptor and its probability saved in a second codebook. The saved codeword is formed by the descriptor and its associated probability. Then, when processing a video for detecting abandoned objects,

Algorithm 2 Second codebook creation algorithm.**Second codebook generation during the second stage training**

The Euclidean distance is used to determine the similarity between the descriptor and the codewords in the codebook

The probability of each volume is calculated in the first training pass

γ is the probability threshold and ϵ_2 is the maximum distance between descriptors that are considered as similar.

```

for All volumes  $\{v_i\}_{i=1}^N$  do
  if  $p_i < \gamma$  then
    if  $\min_j d(v_i, c_j) > \epsilon_2$  then
      Create a new code:  $c_{j+1} \leftarrow v_i$ 
      Create a new code probability:  $p_{c_{j+1}} \leftarrow p_i$ 
    else
      if  $p_i > p_{c_j}$  then
         $p_{c_j} \leftarrow p_i$ 
      end if
    end if
  end if
end for

```

a point is considered anomalous only if it has a probability below the threshold and its descriptor and probability are not close to a codeword in the second codebook. This way the false detections are eliminated while keeping the capacity to detect the anomalies. Algorithm 2 describes the steps to create the new codebook, where γ is the threshold and ϵ_2 is the maximum distance to consider a descriptor as similar to a descriptor from a codebook.

This second codebook must incorporate all the points to be excluded during the anomaly-detection phase for the reference video. This is so because the differences among the probabilities of the volumes tend to be low. So, the training should be done using a reference video including the full surveillance path of the robot. Thus, the number of codewords generated is high, about fifteen new codewords per frame. To speed up the search in the codebook, a paging scheme is employed. Every 50 frames a new page is created, with only the codes generated in these 50 frames. So, the search in the codebook is faster, but the drawback is that the codebook pages of the reference and target video must be synchronized.

In the STC method, the probability threshold directly influences the classification of a volume as anomalous or not. In addition, in this work the threshold is used to choose the points of the second dictionary. Only the points with a probability below the threshold are used to create the second codebook of points of the background with low probability, as explained in the previous section. For all the tests, the probability threshold used had a value γ of 1×10^{-7} .

3.4 Anomaly detection

As in the training stage, the video to be analyzed passes through the same steps of the descriptor creation: allocation of a codeword of the first codebook to each spatio-temporal volume and calculation of the probabilities of these volumes using the pdf of the spatio-temporal arrangement. Then, the second codebook is used to determine the points that should not be considered as anomalous, as they are already part of the background in the reference video. Only points with a probability below a given threshold and not present in the second codebook are considered anomalous. The algorithm is shown in the Algorithm 3.

Algorithm 3 Anomaly Detection algorithm.

The Euclidean distance is used to determine the similarity between the descriptor and the codewords in the codebook

The probability of each volume is calculated in the first training pass

γ is the probability threshold and ϵ_2 is the maximum distance between descriptors that are considered as similar.

μ and ν are the distance and probability tolerance, respectively.

```

for All volumes  $\{v_i\}_{i=1}^N$  do
  if  $p_i < \gamma$  then
    if  $\min_j \text{distance}(v_i, c_j) > \mu\epsilon_2$  then
      if  $p_i > \nu p_{c_j}$  then
        This is an anomalous point. Mark it.
      end if
    end if
  end if
end for

```

3.5 Post-processing

After the detection of the anomalous points, a voting procedure is performed to improve the final result. The reason to execute this vote is that an anomaly may be detected in a certain position of a frame and in the next frame not be present. This is likely to be a false detection because, in most of the cases, the objects do not move so fast that they could disappear from one frame to appear again in another. To perform the voting, the frames are analyzed in groups of 9. Each anomalous object is marked, being identified by its area and centroid position. In order to an object be considered anomalous, it needs to appear in 7 out of 9 consecutive frames. As the object may be moving, a variation of 10 pixels in the centroid position is allowed in any direction.

In some cases, an object can be detected as many separated connected-regions, because of the probability of occurrence of the points inside an object may vary. To lessen the incidence of that effect, a morphological binary closing operation (Gonzalez and Woods 2008) is performed to connect the regions. In this procedure, a round structuring element with radius equal to 20 pixels was used.

4 The VDAO database

The target application of this work is to detect anomalies in videos recorded with moving cameras. To test the limits of the detection in a real scenario a database with videos acquired from a camera mounted on a moving platform and containing abandoned objects was used. In the videos of the database the moving platform surveys an industrial environment by performing translational, back and forth movements along a fixed rail. The camera is arranged so that the image is lateral. Figure 10 shows an example of the type of scene used.

The video database used in the present work is called video database of abandoned objects (VDAO). The database is described in Silva et al. (2014) and available in VDAO (2016). This database, besides containing reference videos without abandoned objects, has videos containing different objects with different colors, shapes and textures (e.g., a shoe, a towel, a box, etc). Figure 11 shows examples of these objects. The position of objects inside the video frame and the time when it is displayed vary from video to video. Moreover, there are variations in brightness between the videos caused by the difference of natural lighting or



Fig. 10 An example of the VDAO database (VDAO 2016). The video was acquired during an inspection of an industrial facility using a camera mounted on a robot that moves along a rail (Color figure online)

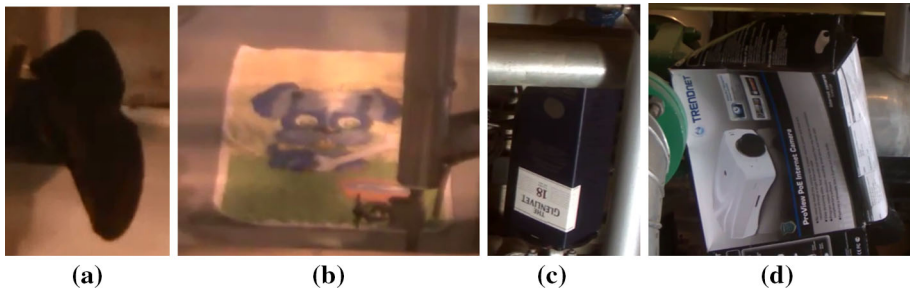


Fig. 11 Examples of abandoned objects present in the database. **a** Shoe. **b** Towel. **c** Dark blue box. **d** Camera box (Color figure online)

the use of an artificial lighting. Another important characteristic of the VDAO database is that the positions of the abandoned objects in all frames of the videos are also provided, in the form of the coordinates of the bounding boxes containing the objects. It is important to highlight that the scenarios presented in this database are common in practical applications of surveillance robots, tending to be quite challenging for video surveillance algorithms.

Although the database videos are composed by about 12,000 frames, due to computational limitations, in our experiments the seven videos were cropped to about 200 frames around the region where there is the object in the scene. Those same video excerpts were used to test all the methods that are compared along this text.

5 The STC-mc algorithm: optimization of its parameters

This section describes the proposed methodology employed in the configuration of the STC-mc algorithm. This is achieved by using the VDAO database to determine the best parameter settings. Initially, the range of variation of each parameter is based on visual inspection of the results of preliminary tests. Then the ground truth of the location of the abandoned objects provided with the VDAO database is employed to automatically compute metrics for the success of the anomaly detection operation. Using these metrics, the values of the parameters are tuned.

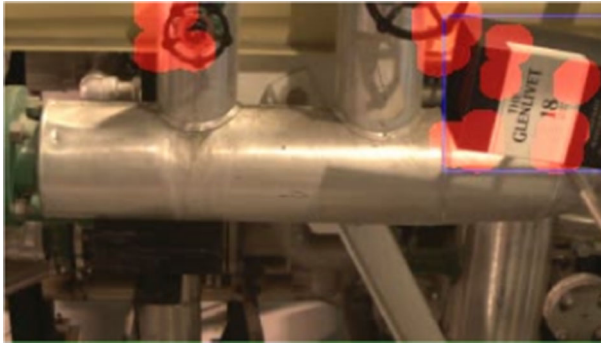


Fig. 12 Example of a marked video and the criterion used for evaluation the detections. The area surrounding the abandoned object is marked with a *blue box*. If a blob contains at least one point inside the *blue box*, it is considered a true positive. A false negative occurs when there is no intersection between any blob and the interior of the box. A blob with all its points outside the *blue box* is considered a false positive. A true negative occurs if there are no blobs with all its points outside the *blue box* (Color figure online)

The ground truth of the VDAO database provides information such as the one depicted in Fig. 12. The area surrounding the abandoned object is marked with a blue box. We refer to a set of contiguous anomalous pixels as a blob. A true positive is an abandoned object with at least one pixel marked as anomalous, that is, a blob with non-monotonic empty intersection with a blue box. A false positive is a blob with no intersection with a blue box. A blue box with no intersection with any blob is considered a false negative. A true negative occurs when no blob has non-empty intersection with a blue box.

After the analysis of the output of the STC-mc algorithm, the number of true positives (TP) and false positives (FP) are used to plot a point of a region of convergence (ROC) curve (Fawcett 2006). Each configuration of parameters generates a point on the curve. In our analysis we consider the best operating point the one closest to the point (1, 0), which corresponds to the point with 100% of true positives and 0% of false positives.

5.1 Parameters of the descriptor and of the codebook from the first stage

There are three main parameters affecting the computation of the spatio-temporal descriptor used in STC-mc: (1) the standard deviation σ of the Gaussian temporal smoothing filter (see Sect. 3.1); (2) the weight of the time derivative λ (Eq. 9); (3) the maximum distance ε_1 above which two codewords are considered to be different (Algorithm 1).

The impulse response of the Gaussian filter is

$$h(n) = \begin{cases} ke^{-\frac{n^2}{2\sigma^2}}, & -2 \leq n \leq 2, \\ 0, & |n| > 2 \end{cases}, \quad (10)$$

where k is such that $\sum_{n=-2}^2 h(n) = 1$.

The set of values of σ used in the tests was {0.8,1.0,1.2}. The weight λ , that is affected by the frame rate of the videos and their resolution, and is used to adjust the sensitivity of the system to the temporal derivative assumed the values {2.5,2.0,1.5}. Lastly, the maximum distance ε_1 is used during the creation of the first codebook. It is the threshold to consider a descriptor as represented by a codeword that is already in the codebook or that a new codeword has to be inserted in the codebook to represent it. The range of variation tested for ε_1 was {600, 700, 800}. As the 3 parameters are interdependent, all their $3 \times 3 \times 3 = 27$

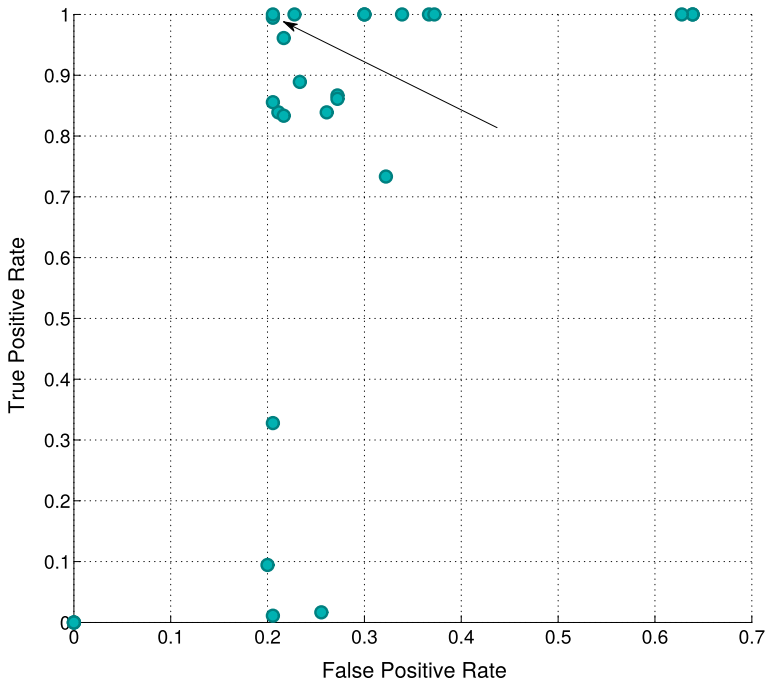


Fig. 13 Scatter results of descriptor parameters estimation tests when the abandoned object is a *whisky box*. The parameters are the filter kernel standard deviation σ , the temporal derivative weight λ and the codebook formation distance ε_1 . The *arrow* indicates the chosen point.

combinations were evaluated, generating a cloud of points instead of a classic ROC curve. In this type of plot, the best operating points are the ones with smallest distance to the point (0, 1) in the plane TP×FP. The result of the test with the video where the abandoned object is a dark blue box, a shoe or a towel are shown in Figs. 13, 14 and 15, respectively.

The parameter set σ , λ and ε_1 that provide the best results for these 3 objects overall is {1.0, 2.0, 700}. For this configuration, the TP×FP values in Figs. 13, 14 and 15 are {0.99×0.20, 0.98×0.01, 1.00×0.00}, respectively.

5.2 Second codebook parameters

To create the codebook from the second stage, it is necessary to determine the value of the maximum distance ε_2 , the distance weight μ and the probability weight ν (see the Algorithms 2 and 3). Like ε_1 for the first codebook, ε_2 is the distance above which a descriptor is considered as not represented by a codeword existent in the second codebook. The set of values used in the tests for ε_2 was {500, 600, 700}. The parameters μ and ν are used during the analysis of a video as described in Algorithms 3. They are necessary because, in a surveillance system, although the video to be analyzed is recorded in roughly the same conditions as the reference video, often there are differences in illumination and even small differences in camera positioning. Therefore, one must introduce some tolerance both when matching a descriptor to a codeword and when applying the probability threshold. In the proposed second-stage dictionary, μ and ν are the distance and probability tolerance, respectively. The set of values of μ employed in the tests was {1.0, 1.05, 1.06, 1.08, 1.10}. The set used

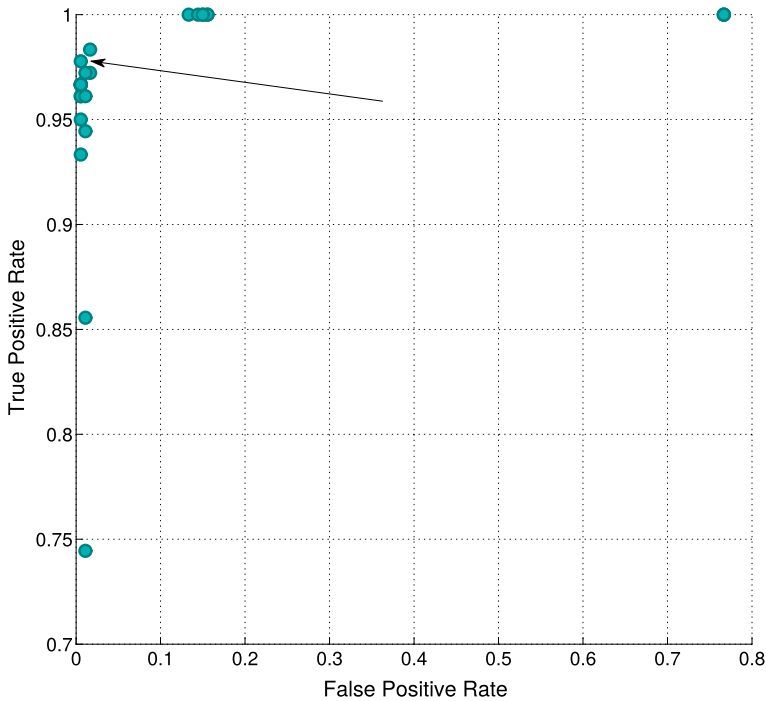


Fig. 14 Scatter results of descriptor parameters estimation tests when the abandoned object is a shoe. The parameters are the filter kernel standard deviation σ , the temporal derivative weight λ and the codebook formation distance ε_1 . The *arrow* indicates the chosen point

for v was $\{2, 4, 6, 7, 8\}$. All the $5 \times 5 = 25$ combinations were simulated. As has been performed for the first pass dictionary (Sect. 5.1), their results generated a cloud of points on the TP \times FP plane. The results for the videos with whisky box, the shoe and the towel are shown in Figs. 16, 17 and 18, respectively.

The set of parameters (ε_2, μ, v) that provide the best overall results (points with smallest distances to the point (0,1) in the TP \times FP plane), taking into account the three sets of points in Figs. 16, 17 and 18, is $\{600, 1.08, 7\}$. For this configuration, the TP \times FP values in the Figs. 16, 17 and 18 are $\{1.00 \times 0.04, 0.90 \times 0.04, 0.92 \times 0.01\}$, respectively.

6 Experimental results

After configuring the parameters of the STC-mc algorithm (as given in Sect. 5), several simulations were performed using the other videos of the VDAO database. This way we could assess both the performance of the STC-mc algorithm and the robustness of the parameter choices for the algorithm by the methodology described in Sect. 5. The parameter set $\{\lambda, \varepsilon_1, \sigma, \gamma, \varepsilon_2, \mu, v\}$ was configured to $\{2, 700, 1, 1 \times 10^{-7}, 600, 1.08, 7\}$.

Figures 19 and 20 show examples of the results obtained. In these figures the abandoned objects are well identified and painted red. Even though not all parts of the abandoned objects were identified, in almost all the cases a reasonable part of the object was properly identified as anomalous. In a real application, where the main objective is to identify the presence of an

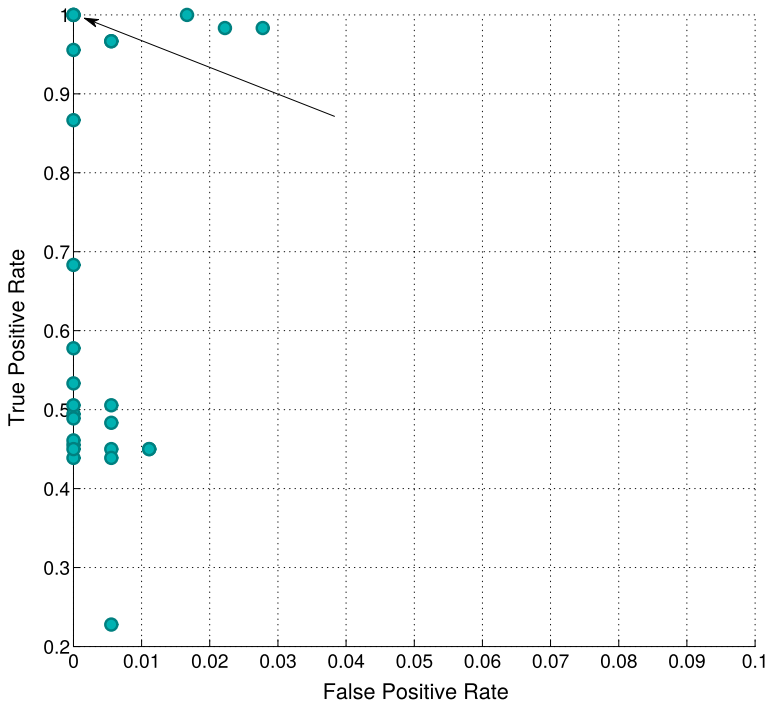


Fig. 15 Scatter results of descriptor parameters estimation tests when the abandoned object is a towel. The parameters are the filter kernel standard deviation σ , the temporal derivative weight λ and the codebook formation distance ε_1 . The *arrow* indicates the chosen point

anomaly, the red spot will be shown in the screen and it will be enough to call the attention of an operator.

Table 1 shows the results of the simulations for seven videos from the VDAO database with abandoned objects. In each video there was an abandoned object: a dark blue box (two different positions), a towel, a shoe, a pink bottle, a camera box and a white jar. An implementation of the BOV algorithm based on the definition presented by Liu and Shah (2008) was used to compare the results. This implementation uses only a dense sampling and a dictionary of BOV to analyze the video. An implementation of the STC method proposed by Roshtkhari and Levine (2013) was also used. Since it uses only the threshold γ to detect the anomalies, it was not possible to find a threshold in which only true positives were present in every frame of the video. Also from Table 1 we note that the original STC method always generates (FP, TP) points that are distant from ideal point (1, 0) for the detection of static objects with a moving camera. The proposed STC-mc algorithm has a good performance detecting the abandoned object in five of the seven videos, with a TP rate greater than 0.9 and a FP rate lower than 0.13. In the videos where the object was not so well detected, the object was very similar to the background and the system was not able to distinguish it. As can be seen in Table 1, when the STC-mc is compared with the BOV implementation, in three of the videos the BOV approach performs slightly better than STC-mc (less than 7%). However, the results of the STC-mc algorithm in these cases are also very good, with a high TP rate and a low FP rate. On the other hand, in four of the seven simulations the STC-mc performs much better than the BOV, indicating a superior performance achieved by the proposed method.

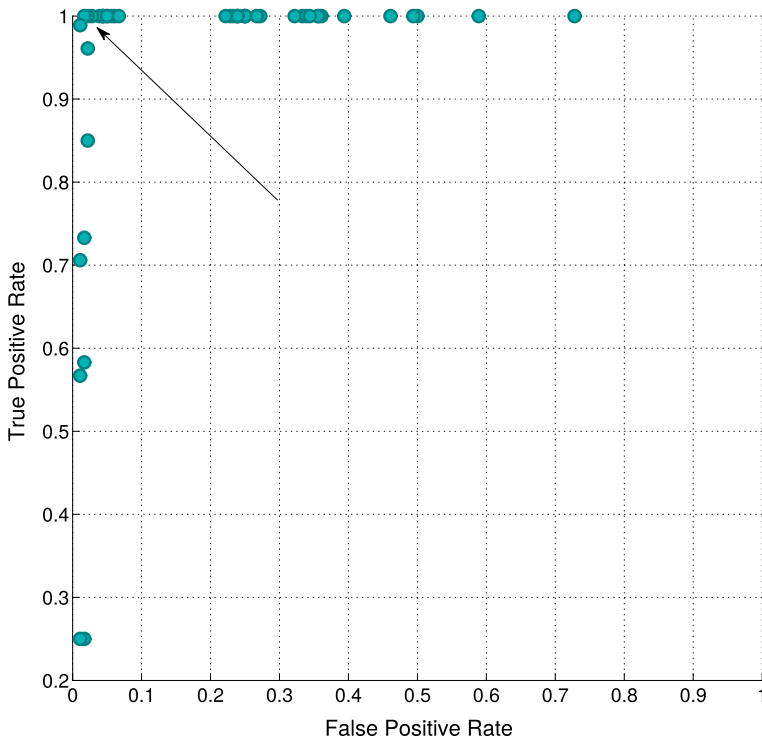


Fig. 16 Scatter results for the parameters of the codebook of the second stage, that is, maximum distance ε_2 , the distance weight μ , and the probability weight ν . The abandoned object was the *whisky box*. The arrow indicates the chosen point

The STC-mc performance was also compared to that of two other methods particularly developed for anomaly detection with a moving camera, the works in [Kong et al. \(2010\)](#) and [Mukojima et al. \(2016\)](#). We have implemented the work of [Kong et al. \(2010\)](#), referred to as DAOMC—Detection of Abandoned Objects with a Moving Camera, replacing the GPS synchronization of the reference and target videos by a manual synchronization. An important observation about our experiments with the DAOMC algorithm is related to the fact that it performs a single frame-by-frame comparison using an NCC window. Therefore, the detection can fail if the object to be detected is much smaller than the NCC window. Then, in all our experiments, the DAOMC algorithm was simulated with using an NCC window that enabled the detection of any object in the VDAO database. The second method used in our comparison is referred to as MCBS—Moving-Camera Background-Subtraction for obstacle detection on railway tracks ([Mukojima et al. 2016](#)). Our implementation of this algorithm replaced the dynamic time-warping (DTW) step with a manual synchronization of the reference and target video sequences. We also removed the post-processing step, which is specific for railway detection problems. The results shown here for the MCBS method employ optimized parameter configuration for the two image similarity metrics used, namely the normalized vector distance (NVD) ([Matsuyama et al. 2000](#)) and radial reach filter (RRF) ([Satoh et al. 2012](#)).

Performance results for the DAOMC and MCBS methods in the same videos considered above are summarized in [Table 2](#). Such results indicate that the STC-mc algorithm, on an

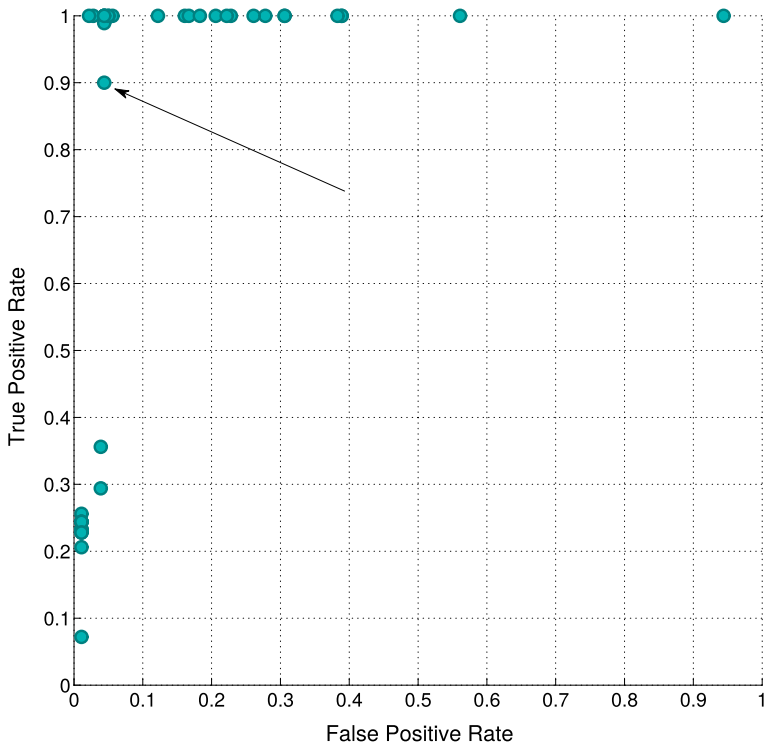


Fig. 17 Scatter results for the parameters of the codebook of the second stage, that is, maximum distance ε_2 , the distance weight μ , and the probability weight ν . The abandoned object was the shoe. The arrow indicates the chosen point

average, has a better performance than the moving-camera state-of-the-art DAOMC and MCBS methods in terms of the DIS metric. DIS is the minimum distance of all operating points to the the ideal behavior, that is $TP = 1$ and $FP = 0$. By analyzing the TP and FP figures, one can see that the STC-mc method has the best $TP \times FP$ trade-off for difficult objects (e.g. Dark blue box 2 or White jar). This is so because the DAOMC and the MCBS methods yield very bad FP results in their failure situations. It is important to note that such problems are minimized in the original DAOMC and MCBS abandoned-object scenarios (road and railroad, respectively), as surrounding cues (horizon and track lines, respectively) remove most FPs in these situations. However, these do not apply in the more general case considered here. Another advantage of the proposed STC-mc algorithm is its relative robustness to the lack of video synchronization, that is provided by the space-time dictionaries. In contrast, both the DAOMC and the MCBS methods require good frame synchronization in order to work properly. Note that in the experiments summarized in Table 2, this advantage is not shown, since the synchronization of the DAOMC and MCBS methods has been performed by hand.

In Table 3 one can see the comparison of the processing times of the STC-mc against the ones of the DAOMC and MCBS anomaly detection methods. The methods were simulated in a computer with an Intel Core i7-4790K processor with 4.00 GHz clock and 32 GB of RAM. The DAOMC algorithm in average proved to be the fastest method and the one with the less variation in the processing time since it performs a frame-by-frame comparison regardless

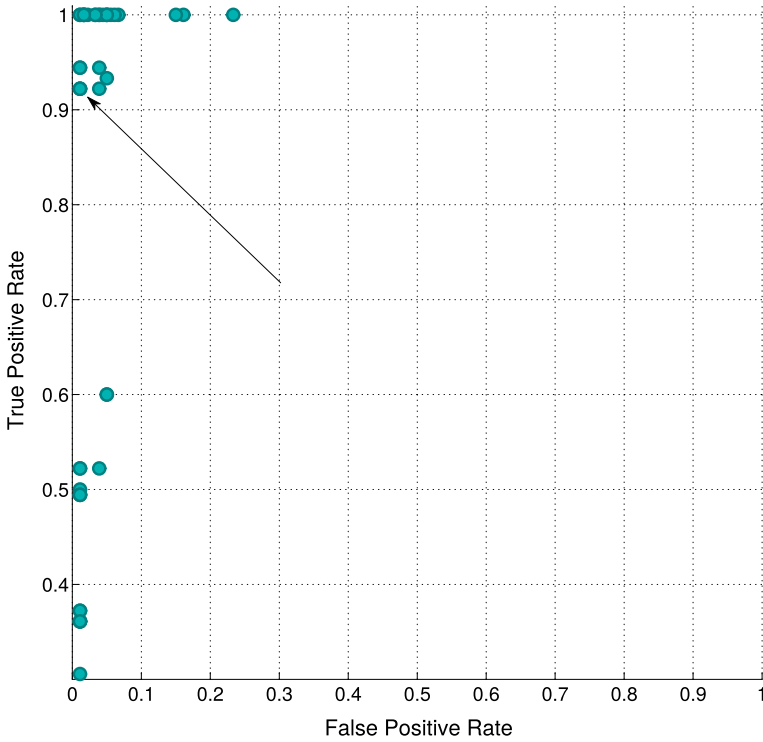


Fig. 18 Scatter results for the parameters of the codebook of the second stage, that is, maximum distance ε_2 , the distance weight μ , and the probability weight ν . The abandoned object was the towel. The arrow indicates the chosen point

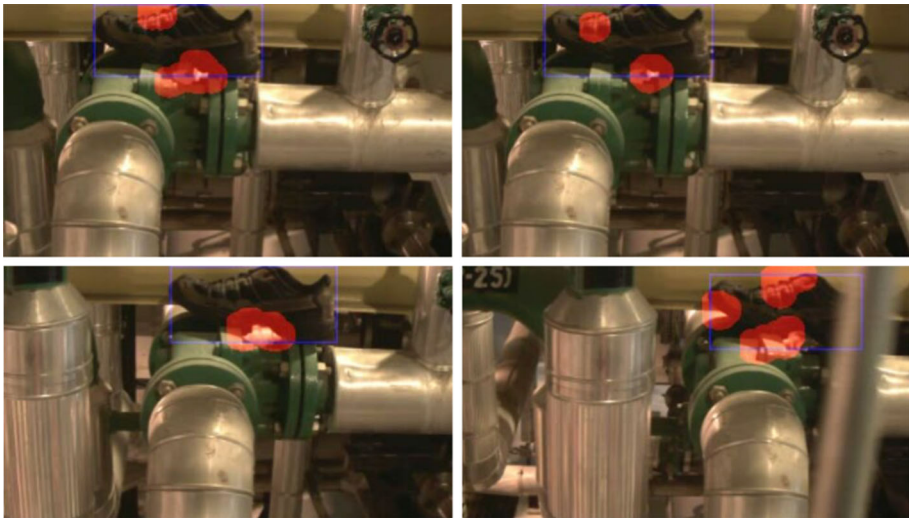


Fig. 19 In this simulation the shoe was the abandoned object and it was detected as an anomaly. The points of the object detected as anomalous are painted in red. The blue box is the bounding box of the shoe as given by the ground truth of the VDAO database (Color figure online)

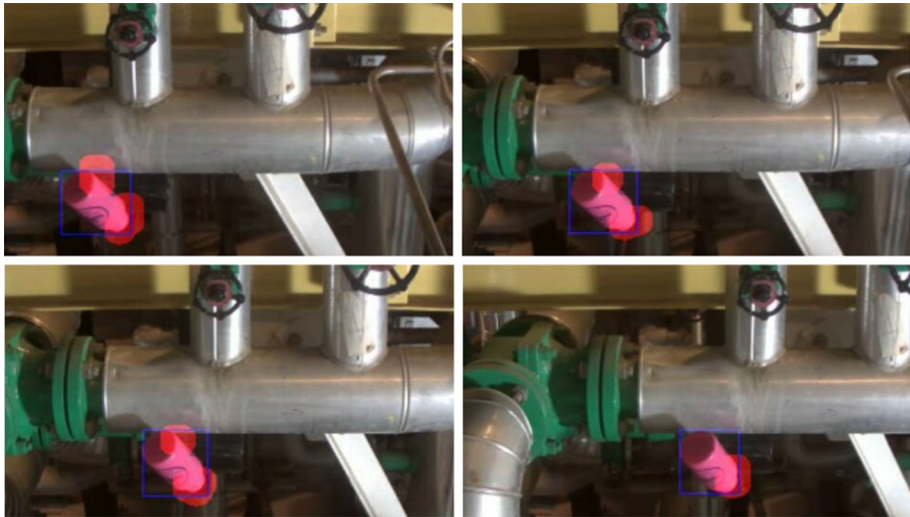


Fig. 20 The *pink bottle* was the abandoned object and it was detected as anomaly. The points of the object detected as anomalous are painted in *red*. The *blue box* is the bounding box of the *pink bottle* as given by the ground truth of the VDAO database (Color figure online)

Table 1 Comparison of the performance of STC-mc against the ones of the BOV and STC methods

Object	STC-mc			BOV			STC		
	TP	FP	DIS	TP	FP	DIS	TP	FP	DIS
Dark blue box 1	1.00	0.04	0.04	1.00	0.22	0.22	0.92	0.82	0.83
Towel	0.92	0.01	0.08	1.00	0.01	0.01	0.93	0.52	0.53
Shoe	0.90	0.04	0.11	1.00	0.04	0.04	0.92	0.94	0.94
Pink bottle	0.99	0.13	0.13	0.99	0.27	0.27	0.00	1.00	1.41
Camera box	1.00	0.03	0.03	1.00	0.01	0.01	1.00	0.79	0.79
Dark blue box 2	0.37	0.42	0.76	0.48	0.64	0.83	0.58	1.00	1.08
White jar	0.29	0.64	0.96	0.01	0.69	1.21	0.85	1.00	1.01

TP is the true positive rate, FP is the false positive rate and DIS is the smallest distance to the point (1, 0) in the TP \times FP plane

of the frame contents. The proposed STC-mc method has comparable complexity to the DAOMC and even outperforms it for two videos. The MCBS method was the slowest of the methods, with a bottleneck in the deep flow computation, responsible for almost 90% of its processing time.

The STC-mc can also be configured to detect anomalies in the static camera case with a moving background. The results obtained with the parameters set $\{\lambda, \varepsilon_1, \sigma, \gamma, \varepsilon_2, \mu, \nu\}$ configured to $\{10, 1.3 \times 10^3, 0.4, 5 \times 10^{-12}, 600, 1.08, 7\}$ are shown in Sect. 2.6. Comparing these with the ones obtained with the original STC in Fig. 6 one can see that STC-mc has a performance as good as the one of the original STC in this case (Fig. 21).

Table 2 Comparison of the performance of STC-mc against the ones of the DAOMC and MCBS methods

Object	STC-mc			DAOMC			MCBS		
	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
Towel	0.92	0.01	0.08	1.00	0.10	0.10	1.00	0.07	0.07
Shoe	0.90	0.04	0.11	1.00	0.04	0.04	1.00	0.28	0.28
Pink bottle	0.99	0.13	0.13	1.00	1.00	1.00	1.00	0.96	0.96
Camera box	1.00	0.03	0.03	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
White jar	0.29	0.64	0.96	1.00	0.10	0.10	1.00	0.99	0.99

TP is the true positive rate, FP is the false positive rate and DIS is the smallest distance to the point (1, 0) in the TP × FP plane

Table 3 Comparison of the processing time in seconds between the anomaly detection methods

Object	STC-mc	DAOMC	MCBS
Dark blue box 1	433	265	50924
Towel	345	280	50403
Shoe	542	293	50427
Pink bottle	415	280	50170
Camera box	448	299	50238
Dark blue box 2	221	289	51740
White jar	248	282	49901

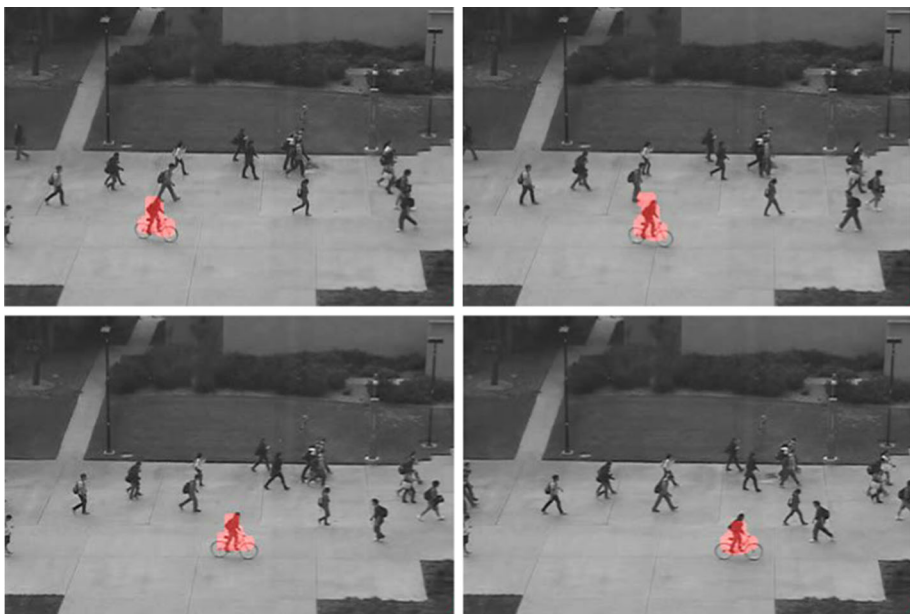


Fig. 21 STC-mc applied to one video of the UCSD database (UCSD 2014). One can see that STC-mc is able to detect anomalous events in the case of a static camera with a moving background (Color figure online)

7 Conclusions

This paper proposed the STC-mc algorithm, a new approach to detect abandoned objects in a cluttered environment, from videos obtained from moving cameras. Although numerous works deal with the detection of abandoned objects, most of them are suitable only to the case of static cameras, and few of them have good results with a moving camera. The proposed STC-mc is based on the same principles as the STC method (Roshtkhari and Levine 2013), which uses dense sampling to break the video in small 3D volumes and calculates the probability of the spatio-temporal arrangements of these volumes. However, the STC-mc algorithm has enhancements that allow it to perform well in the case where the anomalies are abandoned objects and the video is obtained in a cluttered environment using a camera mounted on a moving robot.

This work has three main contributions. One is the use of a new descriptor based on both spatial and temporal gradients. Another is the introduction of a Gaussian filtering to deal with misalignments caused by camera shaking. A third and crucial contribution is the use of a second dictionary that is generated by performing the training in two stages. This second codebook contributes to significantly reduce the number of false detections. It does so by containing codewords representing spatio-temporal compositions that have probabilities in the reference video that are lower than ones commonly associated to anomalies.

The proposed STC-mc algorithm was evaluated by processing several videos of the VDAO database. In most of the cases the STC-mc algorithm was able to detect the abandoned objects with low false positive and high true positive rates. A prior knowledge of the type of event to be detected was not necessary, even in a cluttered environment. In brief, the enhancements proposed to the STC method, that resulted in the STC-mc algorithm, succeeded in the identification of abandoned objects, without background subtraction, motion estimation or tracking. Besides, the STC-mc performs as well as the STC method in the case of a static camera and a moving background. For moving cameras, the STM-mc manages to solve the false detections problem of the original STC algorithm while performing better than a BOV algorithm. In addition, it also achieves comparable or even superior results than the ones of state-of-the-art anomaly-detection methods based on moving cameras, while obviating the need of any frame-by-frame synchronization procedure.

References

- Bay, H., Ess, A., Tuytelaars, T., & Gool, L. V. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3), 346–359.
- Bertini, M., Del Bimbo, A., & Seidenari, L. (2012). Multi-scale and real-time non-parametric approach for anomaly detection and localization. *Computer Vision and Image Understanding*, 116(3), 320–329.
- Bilmes, J. (1998). A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical report, *International Computer Science Institute and Computer Science Division*. University of California at Berkeley.
- Boiman, O., & Irani, M. (2007). Detecting irregularities in images and in video. *International Journal of Computer Vision*, 74(1), 17–31.
- Cheng, G., & Han, J. (2016). A survey on object detection in optical remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 117, 11–28.
- Cheng, G., Zhou, P., & Han, J. (2016). Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12), 7405–7415.
- Cuevas, C., Martínez, R., & García, N. (2016). Detection of stationary foreground objects: A survey. *Computer Vision and Image Understanding*, 152, 41–57.

- de Carvalho, G. H. F. (2015). *Automatic detection of abandoned objects with a moving camera using multiscale video analysis*. D.Sc. thesis, Federal University of Rio de Janeiro, Rio de Janeiro, RJ.
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874.
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.
- Gonzalez, R. C., & Woods, R. E. (2008). *Digital image processing* (3rd ed.). New Jersey: Pearson Prentice Hall.
- Haering, N., Venetianer, P. L., & Lipton, A. (2008). The evolution of video surveillance: An overview. *Machine Vision and Applications*, 19(5–6), 279–290.
- Heijden, F. V. D., Duin, R. P. W., de Ridder, D., & Tax, D. M. J. (2004). *Classification, parameter estimation and state estimation*. West Sussex: Wiley.
- Kong, H., Audibert, J., & Ponce, J. (2010). Detecting abandoned objects with a moving camera. *IEEE Transactions on Image Processing*, 19(8), 2201–2210.
- Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *IEEE conference on computer vision and pattern recognition* (pp. 2169–2178). New York.
- Liu, J., & Shah, M. (2008). Learning human actions via information maximization. In *IEEE conference on computer vision and pattern recognition* (pp. 1–8). Alaska.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110.
- Matsuyama, T., Ohya, T., & Habe, H. (2000). Background subtraction for non-stationary scene. *ACCV 200—Asian conference on computer vision* (pp. 622–667). Taipei.
- Mukojima, H., Deguchi, D., Kawamish, Y., Ide, I., Murase, H., Ukai, M., Nagamine, N., & Nakasone, R. (2016). Moving camera background-subtraction for obstacle detection on railway tracks. In *IEEE international conference on image processing* (pp. 3967–3971). Phoenix.
- Rapantzikos, K., Avrithis, Y., & Kollias, S. (2009). Dense saliency-based spatiotemporal feature points for action recognition. In *IEEE conference on computer vision and pattern recognition* (pp. 1454–1461). Miami.
- Roshkhari, M. J., & Levine, M. D. (2013). An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions. *Computer Vision and Image Understanding*, 117(10), 1436–1452.
- Satoh, Y., Tanahashi, H., Wang, C., Kaneko, S., Niwa, Y., & Yamamoto, K. (2012). Robust event detection by radial reach filter (rrf). In *ICPR 2012—international conference on pattern recognition* (pp. 623–626). Tsukuba Science City.
- Schwartz, O., Hsu, A., & Dayan, P. (2007). Space and time in visual context. *Nature Reviews Neuroscience*, 8(7), 522–535.
- Silva, A. F., Thomaz, L. A., de Carvalho G. H. F., Nakahata, M. T., Jardim, E., Oliveira, J., Silva E. A. B., Netto, S. L., Freitas, G., & Costa, R. R. (2014). An annotated video database for abandoned-object detection in a cluttered environment. In *Proceedings of the 2014 international telecommunications symposium*. Sao Paulo.
- 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.
- UCSD (2014) UCSD anomaly detection dataset. [Online] <http://www.svcl.ucsd.edu/projects/anomaly>.
- VDAO (2016) VDAO—Video database of abandoned objects in a cluttered industrial environment. [Online] <http://www.smt.ufrj.br/~tvdigital/database/objects>.
- Weinzaepfel, P., Revaud, J., Harchaoui, Z., & Schmid, C. (2013). DeepFlow: Large displacement optical flow with deep matching. In *ICCV 2013—IEEE international conference on computer vision* (pp. 1385–1392). Sydney.
- Zhong, H., Shi, J., & Visontai, H. (2004). Detecting unusual activity in video. In *IEEE conference on computer vision and pattern recognition* (pp. 819–826). Washington, DC.
- Zhou, D., Wang, L., Cai, X., & Liu, Y. (2009). Detection of moving targets with a moving camera. In *IEEE international conference on robotics and biomimetics* (pp. 677–681). Guilin.
- Zhou, P., Cheng, G., Liu, Z., Bu, S., & Hu, X. (2016). Weakly supervised target detection in remote sensing images based on transferred deep features and negative bootstrapping. *Multidimensional Systems and Signal Processing*, 27(4), 925–944.



Mateus T. Nakahata was born in Arapongas, Brazil. He received the Electrical Engineering degree from Universidade Federal do Paraná (UFPR), Curitiba, Brazil, in 1998, the M.Sc. degree in Telecommunication from Universidade Federal do Parana (UFPR), Curitiba, Brazil, in 2002 and the D.Sc. degree in Electrical Engineering from Universidade Federal do Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, Brazil, in 2016. His professional experience includes a position as researcher at Instituto de Pesquisa para o Desenvolvimento (LACTEC), in Curitiba, Brazil, from 1998 to 2008, where he was involved in applications of digital signal processing for the electrical industry. Since 2008, he has been a project manager in Petroleo brasileiro (PETROBRAS), Rio de Janeiro, Brazil, working with implementation of telecommunication and CCTV systems. His research interests are digital signal and image processing, signal compression, together with its applications to telecommunication and computer network.



Lucas A. Thomaz was born in Niterói, Brazil. He received the B.Sc. in Electronic and Computer Engineering degree (cum laude) from the Universidade Federal do Rio de Janeiro (UFRJ), Brazil in 2013 and the M.Sc. in Electrical Engineering degree from COPPE/UFRJ in 2015. Since 2015 he pursues his Ph.D. at the Program of Electrical Engineering, at COPPE/UFRJ. 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 Engineering degree in Electronic and Computer Engineering from Universidade Federal do Rio de Janeiro in 2013 and the M.Sc. degree in Electrical Engineering from Universidade Federal do Rio de Janeiro (COPPE/UFRJ) in 2015. Since 2015, he pursues his Ph.D. at Universidade Federal do Rio de Janeiro (COPPE/UFRJ). 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 Instituto Militar de Engenharia (IME), Brazil, in 1984, the M.Sc. degree in Electrical Engineering from Universidade Federal do Rio de Janeiro (COPPE/UFRJ) in 1990, and the Ph.D. degree in Electronics from the University of Essex, England, in 1995. He was with the Department of Electrical Engineering at Instituto Militar de Engenharia, Rio de Janeiro, Brazil in 1987 and 1988, with the Department of Electronics Engineering, UFRJ since 1989 and with the Department of Electrical Engineering, COPPE/UFRJ since 1996. He is co-author of the book “Digital Signal Processing—System Analysis and Design”, published by Cambridge University Press, in 2002, that has also been translated to the Portuguese and Chinese languages, whose second edition has been published in 2010. He has served as associate editor of the IEEE Transactions on Circuits and Systems—I and II, and of Multidimensional, Systems and Signal Processing. He is Deputy Editor-in-Chief of IEEE

Transactions on Circuits and Systems I. He has been a Distinguished Lecturer of the IEEE Circuits and Systems Society. He was Technical Program Co-Chair of ISCAS2011. His research interests lie in the fields of signal and image processing, signal compression, digital TV and pattern recognition, together with its applications to telecommunications and the oil and gas industry.



Sergio L. Netto was born in Rio de Janeiro, Brazil. He received the BSc degree (cum laude) from the Universidade Federal do Rio de Janeiro (UFRJ), Brazil, in 1991, the MSc 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, at Poli/UFRJ, and, since 1998, with the Program of Electrical Engineering, at 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” by Cambridge University Press, 2nd ed., 2010. His research and teaching interests lie in the areas of digital signal processing, speech processing, information theory, and computer vision.