# Anomaly Detection in Moving-Camera Video Sequences Using Principal Subspace Analysis

Lucas A. Thomaz, *Student Member, IEEE*, Eric Jardim, Allan F. da Silva, *Student Member, IEEE*, Eduardo A. B. da Silva, *Senior Member, IEEE*, Sergio L. Netto, *Senior Member, IEEE*, and Hamid Krim, *Fellow, IEEE*

***Abstract*— This paper presents a family of algorithms based on sparse decompositions that detect anomalies in video sequences obtained from slow moving cameras. These algorithms start by computing the union of subspaces that best represents all the frames from a reference (anomaly free) video as a low-rank projection plus a sparse residue. Then, they perform a low-rank representation of a target (possibly anomalous) video by taking advantage of both the union of subspaces and the sparse residue computed from the reference video. Such algorithms provide good detection results while at the same time obviating the need for previous video synchronization. However, this is obtained at the cost of a large computational complexity, which hinders their applicability. Another contribution of this paper approaches this problem by using intrinsic properties of the obtained data representation in order to restrict the search space to the most relevant subspaces, providing computational complexity gains of up to two orders of magnitude. The developed algorithms are shown to cope well with videos acquired in challenging scenarios, as verified by the analysis of 59 videos from the VDAO database that comprises videos with abandoned objects in a cluttered industrial scenario.**

***Index Terms*— Video anomaly detection, sparse representation, object detection, moving camera, subspace recovery.**

## I. Introduction

THE amount of surveillance videos available in private and public facilities has increased exponentially in the past few years, and most probably will keep increasing as surveillance equipment become more accessible and affordable. A report from 2016 [1] estimates the worldwide market for surveillance video equipment around US$43 billions by 2019.

The resulting huge amount of video data creates a problem, as it is unfeasible to humans to watch and analyze properly such content, that is generated on a 24/7 basis. A possible solution for this is the use of automatic surveillance systems that aim at detecting threats, human anomalous activity, presence (or absence) of abandoned (removed) objects, and so on [2]–[7]. Although many works have been developed in this field, there are still many open problems and there is no complete solution for the most general and complex scenarios, such as visually complex or cluttered scenes and dynamic background.

Even though many solutions have been presented for the latter, almost all of them are unable to cope with videos acquired by moving cameras. The use of such cameras tends to increase due to the popularization of moving platforms (e.g. robots, cars, and drones) that perform the surveillance of large areas employing several sensors (e.g. for gases, radiation, etc), that cannot be installed in fixed positions [8]–[11].

The work presented in this paper is focused on methods based on sparse decompositions. Examples of such methods that can cope with surveillance videos acquired from static cameras are discussed in [12] and [13]. They compute low rank representations of the data using subspace decompositions. These approaches, however, besides being restricted to video acquired with static cameras, are not suitable for real-world applications due to the large computational effort required.

We propose to solve the problem of video surveillance using moving cameras by representing the video data as a low rank projection on a union of subspaces (UoS) plus a sparse residue term. We take advantage of the intrinsic structure of the used sparse decomposition in order to detect the anomalies without requiring previous video synchronization. The proposed algorithms project the frames from an anomaly-free reference video in a UoS using a sparse decomposition method and select the best subspaces to reconstruct a possibly anomalous target video. The anomaly detection is performed through the observation of the reconstruction residue, that is, through the target-video part that was not correctly represented by the reference video data [14]. These methods are robust to videos with cluttered backgrounds, such as in industrial plants, as demonstrated by the performed experiments. However, they still suffer from being computationally demanding. To solve this problem, we propose a novel approach that largely reduces the amount of computation in comparison to previous works based on this philosophy.

To properly introduce the proposed techniques, the remainder of this paper is organized as follows: Section II presents some of the related work on moving-camera surveillance,

Manuscript received June 10, 2017; revised August 25, 2017; accepted September 27, 2017. Date of publication October 16, 2017; date of current version February 15, 2018. This work was supported in part by CNPq, in part by FAPERJ, in part by CAPES under Grant 88881.135449/2016-01, and in part by the DOE-National Nuclear Security Administration through CNEC-NCSU under Award DE-NA0002576. This paper was recommended by Associate Editor G. Masera. *(Corresponding author: Lucas A Thomaz.)*

L. A. Thomaz, E. Jardim, A. F. da Silva, E. A. B. da Silva, and S. L. Netto are with the Electrical Engineering Program, COPPE/Universidade Federal do Rio de Janeiro, Rio de Janeiro CEP 21941-972, Brazil (e-mail: lucas.thomaz@smt.ufrj.br; eric.jardim@smt.ufrj.br; allan.freitas@smt.ufrj.br; eduardo@smt.ufrj.br; sergioln@smt.ufrj.br).

H. Krim is with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27606 USA (e-mail: ahk@ncsu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2017.2758379

1549-8328 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

whereas Section III provides a general framework of sparse representation algorithms, introducing the main ideas in the context of well known methods that solve similar sparse representation problems. In Section IV, we describe the proposed surveillance system based on a sparse representation of both reference and target video sequences as well as a detailed analysis of the corresponding computational complexity. Section V is dedicated to the experimental results and comparison with the state-of-the-art methods both in terms of the detection performance and computational complexity. Finally, we present the paper conclusions in Section VI emphasizing main contributions.

## II. Moving-Camera Surveillance Systems

The use of fixed cameras for video anomaly detection usually yields good results as can be seen by the many published works in this field [15]–[19]. In some applications, however, the camera position might suffer small perturbations due to uncontrolled (jitter, wind, vessel movement) or controlled (PTZ, small translation) sources. In these scenarios more powerful methods are required to perform the analysis of the video stream. Often this problem is approached by representing the surveillance videos as matrices and applying sparse decomposition algorithms to it.

One of the most common applications of the latter methods is to detect moving foreground objects. Some of the state-of-the-art techniques designed to deal with this task are briefly discussed here.

The transformed Grassmannian Robust Adaptive Subspace Tracking Algorithmt (t-GRASTA) [20] obtains, from a set of original images, two matrices (low-rank background model and sparse foreground) and a geometric transformation (such as a rotation). In order to do so, it uses an incremental gradient descent (GD) constrained to the Grassmanian manifold of the estimated subspaces.

The Grassmannian Online Subspace Updates with Structured-sparsity (GOSUS) [21] performs the decomposition using an online subspace learning algorithm. It applies a structural restriction to the updates on a Grassmannian manifold based on a group-norm.

The work presented in [22] proposes an online RPCA algorithm that uses geometric transformations for image alignment. Unlike most methods, in [22] these transformations are not applied on the noisy input samples, but only on the recovered samples.

Translational and rotational incremental principal component pursuit (PCP) [23] is a method that aims to processes one frame at a time, avoiding the need for batch processing and yielding a lower memory footprint. It is also capable of dealing with translational and rotational jitter which makes it more robust than its predecessors.

Motion-Aware Graph Regularized RPCA [24] creates a background model by using a modified version of RPCA to generate a low-rank matrix from a set of matrices. In order to do so, an optical flow algorithm is used to estimate the motion, and intra-frame and inter-frame graphs are used to preserve geometric information in the low-rank matrix estimation.

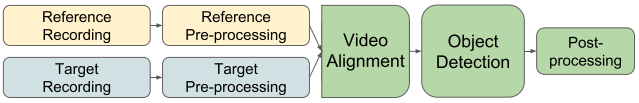


Fig. 1. Traditional framework of moving-camera abandoned object detection including video time and geometric alignment before frame comparison.

The Spatiotemporal Robust Principal Component Analysis (SRPCA) [25] proposes the use of a motion mask that separates the pixels clearly belonging to the foreground. These pixels are labeled as missing data while estimating a temporally smooth background model from the remaining data.

Comprehensive surveys about these low-rank decomposition for forground/background separation methods can be found in [26] and [27] and implementations of the algorithms for several related methods can be found in [28].

In some even more challenging scenarios, the surveillance equipment may be too expensive to be attached in a single fixed position and overlook a single designated part of the environment. In these cases, a possible solution is to attach the equipment to a moving platform enabling one to span a greater surveillance area.

In the cases above, most of the previously discussed methods will not be adequate to obtain the desired results. This is so because, with a camera mounted on a moving platform, usually there are complex, ever-changing backgrounds. These violate an assumption that is common to most of the above methods, the one of small background changes that can be modeled and later subtracted from the videos. In addition, in some cases the anomaly we are trying to detect is caused by the presence of static foreground objects in the scene. This poses an even greater limitation, since without the inputs of other previously recorded videos, one can no longer estimate the background behind the anomalies, thus degrading the performance of most types of foreground/background separation methods.

As depicted in Fig. 1, moving-camera anomaly-detection systems often consider an anomaly-free (as attested by a system operator) reference video and compare it to a target video in search of anomalous situations. Such video-comparison routine is often done on a frame-by-frame basis, thus requiring frame synchronization and geometric alignment of both video sequences. Post-processing is often carried out to take advantage of particular characteristics of each application, such as temporal and spatial consistency of the detection.

A notable attempt to solve the moving-camera anomaly-detection problem was proposed in [29]. In this work a camera mounted on a car searches for abandoned objects on streets. To do so an algorithm similar to that in [30] was used to align the reference and target videos using the GPS signal as an external cue. The frames in this method were geometrically registered using the Random Sampling Consensus (RanSaC) algorithm [31] on Scale-Invariant Feature Transform (SIFT) descriptors [32]. Also, to detect the abandoned objects, the registered frames were compared by computing the Normalized Cross-Correlation (NCC) between the reference and target frames. Despite the method's good performance, the need for an external signal to align reference and target videos limits its usefulness.

The algorithm developed in [33] is able to detect abandoned objects in a heavily cluttered environment in real-time. Video synchronization is performed without the use of any external sensor other than the camera by taking advantage of the a priori knowledge of the camera's linear back-and-forth trajectory. The real-time applicability of this method makes it one of a kind. However, similarly to the method presented in [29] the algorithm's efficiency is also dependent on the correct setup of the NCC window size. Furthermore, the requirement of a specific type of camera movement to perform the video synchronization limits the algorithm applicability in the case of a more general surveillance scenario.

In another recent approach [34] a camera mounted on a train is used to detect the presence of objects across the train path. The alignment and geometric registration techniques (referred to as DeepFlow [35]) used on this method are based on the matching of features extracted with a deep convolutional neural network. This algorithm uses the location of the rails to select the region-of-interest (ROI) in the frame where the algorithm has to search for the anomalous entities, thus avoiding excessive false detections. This method has good performance in the scenario for which it was designed to operate, but has high computational cost due to the DeepFlow-based video alignment. It is also hard to generalize to other surveillance configurations.

More recently, a two-stage dictionary learning approach [36] has been proposed for the analysis of video sequences. It dispenses with the need of motion estimation, tracking or background subtraction. The resulting system considers as anomalies portions of video that are poorly represented by the dictionary. Thanks to the use of a dictionary to represent the target-video images, and unlike most of previous and existing approaches, this algorithm requires neither temporal nor geometric video alignment. The dictionary construction, however, imposes a latency to the system that may not be always tolerable.

In this paper, we describe a new approach for anomaly detection in moving-camera video signals based on a sparse representation of both the reference and target sequences [14], [37]. In the proposed system, the reference video is represented as the combination of a low-rank projection onto a union of subspaces and a sparse residue [13], [38], [39], which are then employed to represent the target video. The residue of this last representation allows one to identify video anomalies in the target video. This scheme obviates the need of temporal alignment between the two video sequences.

## III. Principal Subspace Analysis

When dealing with high-dimensional data one usually wants to find a representation with a reduced dimensionality that allows the data to be analyzed and stored using less resources. A common assumption in those cases is that the data was acquired from a real-world source (e.g. a sensor or a transducer). This implies that it is most likely subjected to noise and other perturbations, which tend to be reduced in the low-dimensional model. In this section we provide a unified framework for some of the main methods used to project high-dimensional data onto subspaces of low dimension, which is known as subspace learning or principal subspace analysis (PSA) [40].

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ be an $m \times n$ data matrix with $\mathbf{x}_i$ comprising $m$-dimensional observations. The projection algorithms model the data as

$$\mathbf{X} = \mathbf{L} + \mathbf{E}, \tag{1}$$

where $\mathbf{L}$ is a low-rank matrix and $\mathbf{E}$ is a sparse residue matrix.

One of the most well-known and widely long used algorithms for this type of analysis is the principal component analysis (PCA) [41], which employs the singular value decomposition (SVD) to find out the orthogonal basis that supports the low-dimensional data subspace, while casting the remaining noisy components to the residue matrix. This approach is able to find the optimal subspace that minimizes the projection error of the columns of $\mathbf{X}$ and may be expressed as

$$\min_{\mathbf{L},\mathbf{E}} ||\mathbf{E}||_{\mathrm{F}} \quad \text{s.t.} \begin{cases} \mathbf{X} = \mathbf{L} + \mathbf{E} \\ \mathrm{rank}(\mathbf{L}) \leq r, \end{cases} \tag{2}$$

where $||.||_{\mathrm{F}}$ denotes the Frobenius [42] norm and $r$ is the maximum rank of matrix $\mathbf{L}$. The PCA, however, is only able to cope with small corruptions in the original data, since large corruption levels modify the subspace support vectors significantly, compromising the resulting data decomposition. Also, the maximum rank of the $\mathbf{L}$ matrix must be known a priori, thus requiring some previous knowledge about the data.

The so-called robust PCA (RPCA) [12] is a refined version of the PCA algorithm that is able to recover a low-rank matrix $\mathbf{L}$ even when the original data matrix $\mathbf{X}$ includes outliers (heavy tail noise). Note that formulation of RPCA assumes the rank ($r$) unknown, and hence an intrinsic property of the underlying model to be unveiled. Mathematically the formulation of RPCA may be written as

$$\min_{\mathbf{L},\mathbf{E}} \ \mathrm{rank}(\mathbf{L}) + \lambda ||\mathbf{E}||_0 \quad \text{s.t.} \ \ \mathbf{X} = \mathbf{L} + \mathbf{E}, \tag{3}$$

where $||.||_0$ is the $l_0$-norm (number of non-zero entries in the matrix) and $\lambda$ is a weighting parameter. Although this problem formulation is very simple and effective, it is an intractable NP-hard problem that cannot be solved effectively for large data sizes. A relaxed version is often used [12],

$$\min_{\mathbf{L},\mathbf{E}} \ ||\mathbf{L}||_* + \lambda ||\mathbf{E}||_1 \quad \text{s.t.} \ \mathbf{X} = \mathbf{L} + \mathbf{E}, \tag{4}$$

where $||.||_*$ is the nuclear norm (defined as $||\mathbf{A}||_* = \mathrm{tr}(\sqrt{\mathbf{A}^{\mathrm{H}}\mathbf{A}})$, with $\mathbf{A}^{\mathrm{H}}$ denoting the conjugate transpose of $\mathbf{A}$) and $||\mathbf{A}||_1$ is the sum of the absolute values of all the entries of $\mathbf{A}$.

Both PCA and RPCA are able to project the data onto a single subspace. When the data matrix is better interpreted by the projection onto a union of subspaces of lower dimensions, one may consider the Robust Subspace Recovery (RoSuRe) algorithm proposed in [13] and [39]. In this formulation, one considers the UoS $\mathcal{S} = \cup_{j=1}^{J} \mathcal{S}^{(j)}$ with $\mathbf{L}$ being a matrix whose

columns are uniformly sampled from $\mathcal{S}$. We group all the samples from the same subspace $\mathcal{S}^{(j)}$ into matrix $\mathbf{L}^{(j)}$ so that

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}^{(1)} & \mathbf{L}^{(2)} & \cdots & \mathbf{L}^{(J)} \end{bmatrix}. \tag{5}$$

With sufficient sampling density, every column $\mathbf{l}_k^{(j)}$ of $\mathbf{L}^{(j)}$ can be represented by a linear combination of the other columns $\mathbf{l}_i^{(j)}$, $i \neq k$ from the same subspace. In this case, one can say that the set of columns of $\mathbf{L}^{(j)}$ is self-representative, and it is possible to state that

$$\mathbf{L}^{(j)} = \mathbf{L}^{(j)}\mathbf{W}^{(j)}, \tag{6}$$

where $\mathbf{W}_{k,k}^{(j)} = 0$.

As a result, from Eq. (5) one can write that $\mathbf{L} = \mathbf{LW}$, with

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}^{(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{W}^{(2)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{W}^{(J)} \end{bmatrix}, \tag{7}$$

where, from Eq. (6), $\mathbf{W}_{k,k}^{(j)} = 0$, $j = 1, \ldots, J$. Note that by observing the underlying structure from $\mathbf{W}$ it is possible to infer the subspace structure from $\mathbf{L}$, which is said to be blockwise low-rank as induced by $\mathbf{W}$.

Let now $\mathbf{X}$ be such that it can be represented as an element belonging to the UoS $\mathcal{S}$ added to a sparse residue $\mathbf{E}$. This is equivalent to stating it can be decomposed as

$$\mathbf{X} = \mathbf{LW} + \mathbf{E}, \tag{8}$$

where, from Eq. (7), $\mathbf{W}$ is blockwise diagonal with $\mathbf{W}_{k,k} = 0$ for all $k$, $\mathbf{L}$ is blockwise low-rank, and $\mathbf{E}$ is sparse.

The RoSuRe method assumes sparsity both on $\mathbf{W}$ (due to its structure) and $\mathbf{E}$ (as it is considered that each $\mathbf{E}_i$ is sparse). To perform the decomposition and assure the above constraints, the method solves the following optimization problem:

$$\min_{\mathbf{W},\mathbf{E}} ||\mathbf{W}||_0 + \lambda||\mathbf{E}||_0, \quad \text{s.t.} \begin{cases} \mathbf{X} = \mathbf{L} + \mathbf{E} \\ \mathbf{LW} = \mathbf{L} \\ \mathbf{W}_{ii} = 0, \ \forall i \end{cases}, \tag{9}$$

where $||\cdot||_0$ represents the number of non-zero entries on the matrix. As this is a hard non-convex optimization problem, [13] proposes solving a relaxation of it given by

$$\min_{\mathbf{W},\mathbf{E}} ||\mathbf{W}||_1 + \lambda||\mathbf{E}||_1, \quad \text{s.t.} \begin{cases} \mathbf{X} = \mathbf{L} + \mathbf{E} \\ \mathbf{LW} = \mathbf{L} \\ \mathbf{W}_{ii} = 0, \quad \forall i. \end{cases} \tag{10}$$

The optimization proposed in Eq. (10) can be solved with the use of Algorithm 1. In this and in the subsequent algorithms, the variable $\mu_k$ is the augmented Lagrange multiplier, $\rho$ is the step used to update $\mu_k$, $\eta_1 \geq ||\mathbf{L}||_2^2$ and $\eta_2 \geq ||\hat{\mathbf{W}}||_2^2$ are normalizing weights, $\tau_\alpha(\cdot)$ is the soft-thresholding operator for the Augmented Lagrangian Multiplier, defined as [43]

$$\tau_\alpha(x) = \begin{cases} x - \alpha, & x \geq \alpha \\ 0, & |x| \leq \alpha \\ x + \alpha, & x \leq -\alpha. \end{cases} \tag{11}$$

Further details can be found on [13].

**Algorithm 1** RoSuRe

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, $\lambda$, $\rho > 1$, $\eta_1$, $\eta_2$, $\mu_0$, $\mathbf{W}_0 = \hat{\mathbf{W}}_0 = \mathbf{E}_0 = \mathbf{Y}_0 = \mathbf{0}$.
**while** not converged **do**
  Update $\mathbf{W}$ by linearized soft-thresholding:
  $\mathbf{L}_{k+1} = \mathbf{X} - \mathbf{E}_k$
  $\mathbf{W}_{k+1} = \tau_{\frac{\lambda}{\mu\eta_1}} \left( \mathbf{W}_k - \frac{1}{\eta_1}\mathbf{L}_{k+1}^{\mathrm{T}} \left( \mathbf{L}_{k+1}\hat{\mathbf{W}}_k - \frac{\mathbf{Y}_k}{\mu_k} \right) \right)$
  $\mathbf{W}_{k+1}^{ii} = 0$
  Update $\mathbf{E}$ by linearized soft-thresholding:
  $\hat{\mathbf{W}}_{k+1} = \mathbf{I} - \mathbf{W}_k$
  $\mathbf{E}_{k+1} = \tau_{\frac{1}{\mu\eta_2}} \left( \mathbf{E}_k + \frac{1}{\eta_2}(\mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1} - \frac{\mathbf{Y}_k}{\mu_k})\hat{\mathbf{W}}_{k+1}^{\mathrm{T}} \right)$
  Update the Lagrange multiplier Y and the augmented Lagrange multiplier $\mu_k$
  $\mathbf{Y}_{k+1} = \mathbf{Y}_k + \mu_k (\mathbf{L}_{k+1}\mathbf{W}_{k+1} - \mathbf{L}_{k+1})$
  $\mu_{k+1} = \rho\mu_k$
**end while**

The RoSuRe algorithm was proven [39] to work properly on synthetic and real data created by randomly sampling vectors from UoS and adding sparse corrupting noise with different signal-to-noise ratios (SNR).

## IV. Moving-Camera Video Anomaly Detection Using RoSuRe Decomposition

Principal subspace analysis (PSA) methods can be used to solve many practical problems. If, for instance, one assumes a slowly moving camera, then consecutive frames of a given reference video $\mathbf{X}_r$ share approximately the same low-rank RoSuRe representation [13], [39] allowing one to write that

$$\mathbf{X}_r = \mathbf{L}_r\mathbf{W}_r + \mathbf{E}_r, \tag{12}$$

$$\mathbf{E}_r = \mathbf{X}_r - \mathbf{L}_r, \tag{13}$$

where $\mathbf{L}_r$ is the low-rank[1] representation of $\mathbf{X}_r$ and $\mathbf{E}_r$ is its sparse complement. Note that Eqs. (12) and (13) imply that $\mathbf{L}_r\mathbf{W}_r = \mathbf{L}_r$. The corresponding optimization problem then becomes

$$\min_{\mathbf{W}_r,\mathbf{E}_r} ||\mathbf{W}_r||_1 + \lambda||\mathbf{E}_r||_1, \quad \text{s.t.} \begin{cases} \mathbf{X}_r = \mathbf{L}_r + \mathbf{E}_r \\ \mathbf{L}_r\mathbf{W}_r = \mathbf{L}_r \\ \mathbf{W}_{r_{ii}} = 0, \quad \forall i. \end{cases} \tag{14}$$

In the absence of any anomalous content, the corresponding frames in both the reference and target videos in the surveillance system depicted in Fig. 1 share the same low-rank representation. Therefore, one can use the low-rank representation $\mathbf{L}_r$ of $\mathbf{X}_r$ to represent the target video $\mathbf{X}_t$ such that

$$\mathbf{X}_t = \mathbf{L}_r\mathbf{W}_t + \mathbf{E}_t, \tag{15}$$

[1]The self-representative matrix $\mathbf{L}_r$ is guaranteed to be low-rank for a single subspace. For a UoS, as presented in this case, it is usually low-rank, but there may be cases where the construction of a specific UoS may not lead to a low-rank matrix $\mathbf{L}_r$. Nevertheless, as for making the notation of the methodology compatible with that of previous works we will refer to $\mathbf{L}_r$ as either "low-rank" or "self-representative" matrix interchangeably.

with $\mathbf{W}_t$ and $\mathbf{E}_t$ both being sparse matrices, to which the corresponding optimization is

$$\min_{\mathbf{W}_t,\mathbf{E}_t} ||\mathbf{W}_t||_1 + \lambda||\mathbf{E}_t||_1, \quad \text{s.t.} \quad \mathbf{L}_r\mathbf{W}_t = \mathbf{X}_t - \mathbf{E}_t. \tag{16}$$

By modifying the original RoSuRe algorithm [13], the optimization problem in Eq. (16) can be solved as summarized in Algorithm 2.

**Algorithm 2** Sparse Representation of $\mathbf{X}$ Given the Low-Rank Representation $\mathbf{L}$

Input: $\mathbf{L}$, $\mathbf{X}$, $\lambda$,$\rho > 1$, $\eta_1$, $\eta_2$, $\mu_0$, $\mathbf{W}_0 = \mathbf{E}_0 = \mathbf{Y}_0 = \mathbf{0}$.
**while** not converged **do**
$\mathbf{L}'_{k+1} = \mathbf{X} - \mathbf{E}_k$
$\mathbf{W}_{k+1} = \tau_{\frac{\lambda}{\mu\eta_1}}\left(\mathbf{W}_k - \frac{1}{\eta_1}\mathbf{L}^{\mathrm{T}}\left(\mathbf{L}\mathbf{W}_k - \mathbf{L}'_{k+1} + \frac{\mathbf{Y}_k}{\mu_k}\right)\right)$
$\mathbf{E}_{k+1} = \tau_{\frac{1}{\mu\eta_2}}\left(\mathbf{E}_k - \frac{1}{\eta_2}\left(\mathbf{L}\mathbf{W}_{k+1} - \mathbf{L}'_{k+1} + \frac{\mathbf{Y}_k}{\mu_k}\right)\right)$
$\mathbf{Y}_{k+1} = \mathbf{Y}_k + \mu_k\left(\mathbf{L}\mathbf{W}_{k+1} - \mathbf{L}'_{k+1}\right)$
$\mu_{k+1} = \rho\mu_k$
**end while**

Solving the problem in Eq. (16), all the anomalous information in $\mathbf{X}_t$ that could not be represented from $\mathbf{L}_r\mathbf{W}_t$ are cast upon $\mathbf{E}_t$. Actually, there are in $\mathbf{E}_t$ other artifacts (such as high-frequency components not representable by the low-rank matrix $\mathbf{L}_r$) that are not related to the anomalies of interest. Those artifacts, however, are indeed supposed to be present in matrix $\mathbf{E}_r$. Therefore, one can remove these artifacts from $\mathbf{E}_t$ by performing an additional decomposition of this matrix using $\mathbf{E}_r$ as its low-rank component, as given by

$$\mathbf{E}_t = \mathbf{E}_r\mathbf{W}_e + \mathbf{E}_e, \tag{17}$$

such that the final residue matrix $\mathbf{E}_e$ contain only the anomalies of interest in the target video. To allow such representation, one has to perform the following optimization

$$\min_{\mathbf{W}_e,\mathbf{E}_e} ||\mathbf{W}_e||_1 + \lambda||\mathbf{E}_e||_1, \quad \text{s.t.} \quad \mathbf{E}_r\mathbf{W}_e = \mathbf{E}_t - \mathbf{E}_e \tag{18}$$

A summarized version of the complete moving-camera RoSuRe (mcRoSuRe) algorithm is presented in Algorithm 3.

**Algorithm 3** Moving-Camera RoSuRe Algorithm

**Require:** $\mathbf{X}_r$, $\mathbf{X}_t$
$\min_{\mathbf{W}_r,\mathbf{E}_r} ||\mathbf{W}_r||_1 + \lambda||\mathbf{E}_r||_1$, s.t. $\mathbf{X}_\mathrm{r} = \mathbf{L}_\mathrm{r} + \mathbf{E}_\mathrm{r}$, $\mathbf{L}_r\mathbf{W}_r = \mathbf{L}_r$, $\mathbf{W}_{r_{ii}} = 0$
$\min_{\mathbf{W}_t,\mathbf{E}_t} ||\mathbf{W}_t||_1 + \lambda||\mathbf{E}_t||_1$, s.t. $\mathbf{L}_\mathrm{r}\mathbf{W}_\mathrm{t} = \mathbf{X}_\mathrm{t} - \mathbf{E}_\mathrm{t}$
$\min_{\mathbf{W}_e,\mathbf{E}_e} ||\mathbf{W}_e||_1 + \lambda||\mathbf{E}_e||_1$, s.t. $\mathbf{E}_\mathrm{r}\mathbf{W}_\mathrm{e} = \mathbf{E}_\mathrm{t} - \mathbf{E}_\mathrm{e}$

### *A. Accelerated Versions of mcRoSuRe Anomaly-Detection Algorithm*

The mcRoSuRe algorithm shows great performance in the detection of abandoned objects in a cluttered environment, with a good detection performance and a reduced false-positive rate. However, the algorithm is computationally intensive and is therefore not suited for real-time applications. In fact, the computational complexity of the mcRoSuRe algorithm increases significantly with the size of the videos being analyzed (see Subsection IV-B for a precise analysis). This explains the small video excerpts (70-frame long videos of 320 × 180-pixel frames) processed in [14]. To allow the reduction on the execution time of the algorithm, one may take advantage of some of its intrinsic properties concerning the resulting data representation. In this subsection new accelerating techniques that benefit from this innate representation (including the ones introduced in [44]) and modify the original method are discussed.

The original mcRoSuRe formulation does not require a precise frame-by-frame synchronization of the reference and target videos, but only that the area covered by the target video is contained within the area covered by the reference video excerpt. This is clear from the analysis of Eq. (12), where target-video data matrix $\mathbf{X}_t$ can be reconstructed by $\mathbf{L}_r$, the low-rank component of the reference video, up to a sparse error $\mathbf{E}_t$. If one could reduce the number of columns of $\mathbf{L}_r$ to include only those corresponding to the exact portion of the target video under analysis, great computational savings could be obtained. This is the same as saying that the UoS search space in the optimization problem described in Eq. (16) is restricted to a limited number of relevant subspaces.

One way of selecting these reference frames of interest is to observe the resulting $\mathbf{W}_t$ matrix in Eq. (15). This requires, however, the computationally expensive implementation of the first two steps of the mcRoSuRe algorithm described in Eqs. (14) and (16), that are detailed in Algorithm 3. One way to avoid this issue is to precompute $\mathbf{W}_t$ by representing the frames from the target video not as a combination of the low-rank representations of the reference frames $\mathbf{L}_r$, but as a combination of the actual reference frames $\mathbf{X}_r$. This proposition allows the construction of a version of the $\mathbf{W}_t$ matrix without the need to find the low-rank representation of the $\mathbf{X}_r$ matrix. This is the most costly step of the mcRoSuRe algorithm as will be shown later in the experimental results section. To perform this precomputation step one should compute the decomposition below [44]

$$\mathbf{X}_t = \mathbf{X}_r\mathbf{W}_t + \mathbf{E}_t. \tag{19}$$

This new added step requires solving the optimization problem defined by

$$\min_{\mathbf{W}_t,\mathbf{E}_t} ||\mathbf{W}_t||_1 + \lambda||\mathbf{E}_t||_1, \quad \text{s.t. } \mathbf{X}_t = \mathbf{X}_r\mathbf{W}_t + \mathbf{E}_t, \tag{20}$$

whose implementation is summarized in Algorithm 4.

**Algorithm 4** Decomposition of $\mathbf{X}_t$ Using $\mathbf{X}_r$ Instead of $\mathbf{L}_r$

Input: $\mathbf{X}'_r$, $\mathbf{X}_t$, $\lambda$, $\rho > 1$, $\eta_1$, $\eta_2$, $\mu_0$, $\mathbf{W}_0 = \mathbf{E}_0 = \mathbf{Y}_0 = \mathbf{0}$.
**while** not converged **do**
$\mathbf{X}'_{r(k+1)} = \mathbf{X}_t - \mathbf{E}_k$
$\mathbf{W}_{k+1} = \tau_{\frac{\lambda}{\mu\eta_1}}\left(\mathbf{W}_k - \frac{1}{\eta_1}\mathbf{X}_r^{\mathrm{T}}\left(\mathbf{X}_r\mathbf{W}_k - \mathbf{X}'_{r(k+1)} + \frac{\mathbf{Y}_k}{\mu_k}\right)\right)$
$\mathbf{E}_{k+1} = \tau_{\frac{1}{\mu\eta_2}}\left(\mathbf{E}_k - \frac{1}{\eta_2}\left(\mathbf{X}_\mathbf{r}\mathbf{W}_{k+1} - \mathbf{X}'_{r(k+1)} + \frac{\mathbf{Y}_k}{\mu_k}\right)\right)$
$\mathbf{Y}_{k+1} = \mathbf{Y}_k + \mu_k\left(\mathbf{X}_r\mathbf{W}_{k+1} - \mathbf{X}'_{r(k+1)}\right)$
$\mu_{k+1} = \rho\mu_k$
**end while**

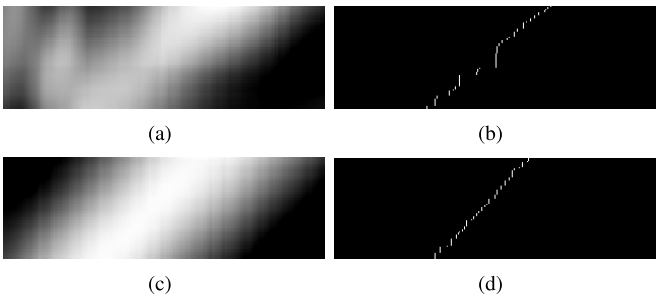

Fig. 2. Results for $\mathbf{W}_t$ matrix being used to localize the important corresponding frames of $\mathbf{X}_r$ (brighter pixels denote higher values in the matrices). The vertical dimension corresponds to the target video frames and the horizontal dimension to the reference video frames: (a) $\mathbf{W}_t$ matrix from Eq. (15); (b) Columnwise maximum of $\mathbf{W}_t$ matrix from Eq. (15); (c) $\mathbf{W}_t$ matrix from Eq. (19); (d) Columnwise maximum of $\mathbf{W}_t$ matrix from Eq. (19).

Fig. 3. Example of resulting $\mathbf{W}_t$ matrices from Eq. (19) using: (a) complete reference data matrix $\mathbf{X}_r$; (b) downsampled-in-time reference matrix $\mathbf{X}_r^{\mathrm{ds}}$.

A direct comparison between the corresponding $\mathbf{W}_t$ matrices generated by Eqs. (15) and (19), respectively, is shown in Fig. 2, where one can readily observe that Eq. (19) yields less spread and more precise results with respect to localization of the important frames. That behaviour most likely comes from the fact that the high-frequency components within $\mathbf{X}_r$ (which are absent from $\mathbf{L}_r$) are crucial in selecting the proper frame of $\mathbf{X}_r$ to represent the corresponding frame in $\mathbf{X}_t$.

By detecting the columnwise maximum of the $\mathbf{W}_t$ matrices, as depicted in the right-hand side of Fig. 2, one can find the direct correspondence between the frames of $\mathbf{X}_r$ and $\mathbf{X}_t$. Selecting from $\mathbf{X}_r$ only the frames that correspond to the portion of the target video $\mathbf{X}_t$ being analyzed one can execute the optimization steps represented by Eqs. (14), (16), and (18) replacing $\mathbf{X}_r$ by a much smaller $\mathbf{X}'_r$ matrix, thus reducing the computational cost associated to the resulting algorithm. This algorithm is referred to as mcRoSuRe with Temporal Alignment (mcRoSuRe-TA) and was first introduced in [44].

Furthermore, one can use the above formulation to further reduce the computation complexity. This can be achieved by performing a uniform temporal subsampling of the original reference video, yielding a smaller, yet representative, reference data matrix $\mathbf{X}_r^{\mathrm{ds}}$. If one observes the $\mathbf{W}_t$ matrices obtained by the decomposition performed by Eq. (16) with the original $\mathbf{X}_r$ and with $\mathbf{X}_r^{\mathrm{ds}}$ it is possible to observe that the width of $\mathbf{W}_t$ matrix computed with $\mathbf{X}_r^{\mathrm{ds}}$ is very much reduced in comparison with that obtained with the original $\mathbf{X}_r$ matrix. Nevertheless, the interval that relates to the frames of the target video is still clear, allowing a precise selection of the reference frames used to decompose the target video. Fig. 3 shows an example of the $\mathbf{W}_t$ matrices generated using the original $\mathbf{X}_r$ and decimated-in-time $\mathbf{X}_r^{\mathrm{ds}}$.

From this figure, one can readily see the size discrepancy between the two approaches, which translates in a much reduced computational effort for the latter. Note that $\mathbf{X}'_r$ corresponds to an interval of $\mathbf{X}_r$; $\mathbf{X}_r^{\mathrm{ds}}$ is only used to determine the limits of this interval.

TABLE I

VARIABLES RELATED TO THE ASSESSMENT OF THE COMPUTATIONAL COMPLEXITY OF THE ALGORITHMS

| Variable | Related quantity |
|---|---|
| $P$ | Total number of pixels in each frame |
| $N_r$ | Number of columns (frames) in $\mathbf{X}_r$ |
| $N_t$ | Number of columns (frames) in $\mathbf{X}_t$ |
| $N_r^{\mathrm{ds}}$ | Number of columns (frames) in $\mathbf{X}_r^{\mathrm{ds}}$ |
| $N'_r$ | Number of columns (frames) in $\mathbf{X}'_r$ |

With the addition of the proposed pre-processing steps, the accelerated version of the mcRoSuRe approach becomes as summarized in Algorithm 5 and is called mcRoSuRe Accelerated (mcRoSuRe-A).

**Algorithm 5** mcRoSuRe - Accelerated

Downsample reference video to create $\mathbf{X}_r^{\mathrm{ds}}$.
$\min_{\mathbf{W}_t,\mathbf{E}_t} ||\mathbf{W}_t||_1 + \lambda||\mathbf{E}_t||_1$, s.t. $\mathbf{X}_{\mathrm{t}} = \mathbf{X}_{\mathrm{r}}^{\mathrm{ds}}\mathbf{W}_{\mathrm{t}} + \mathbf{E}_{\mathrm{t}}$,
Crop reference frames of interest based on $\mathbf{W}_t$ matrix. Create $\mathbf{X}'_r$.
$\min_{\mathbf{W}'_r,\mathbf{E}'_r} ||\mathbf{W}'_r||_1 + \lambda||\mathbf{E}'_r||_1$, s.t. $\mathbf{X}'_{\mathrm{r}} = \mathbf{L}'_{\mathrm{r}} + \mathbf{E}'_{\mathrm{r}}$, $\mathbf{L}'_r\mathbf{W}'_r = \mathbf{L}'_r$, $\mathbf{W}'_{r_{ii}} = 0$
$\min_{\mathbf{W}'_t,\mathbf{E}'_t} ||\mathbf{W}'_t||_1 + \lambda||\mathbf{E}'_t||_1$, s.t. $\mathbf{X}'_{\mathrm{r}}\mathbf{W}'_{\mathrm{t}} = \mathbf{X}_{\mathrm{t}} - \mathbf{E}'_{\mathrm{t}}$
$\min_{\mathbf{W}_e,\mathbf{E}_e} ||\mathbf{W}_e||_1 + \lambda||\mathbf{E}_e||_1$, s.t. $\mathbf{E}'_{\mathrm{r}}\mathbf{W}_{\mathrm{e}} = \mathbf{E}'_{\mathrm{t}} - \mathbf{E}'_{\mathrm{e}}$

### *B. Computational Complexity Analysis*

We now consider the number of arithmetic operations required to implement the different versions of the mcRoSuRe algorithm discussed in Section IV. For the calculation of the number of computations, the numbers of additions and multiplications were obtained from Algorithms 1, 2, and 4. For this analysis, let $N_r$ and $N_t$ be the numbers of $R \times C$-pixel frames in the reference and target videos, respectively, and let $P = RC$ be the number of pixels per frame, as indicated in Table I.

The RoSuRe method, described in Algorithm 1, operates on $N_r \times P$ matrices, where each iteration requires

$$A(N_r, P) = 2N_r^2 + 5PN_r \tag{21}$$

additions and

$$M(N_r, P) = 4PN_r^2 + 2N_r^2 + 3PN_r + 7 \tag{22}$$

multiplications, which, in practice, is dominated by the $\mathcal{O}(PN_r^2)$ term [13].

The more computationally intensive mcRoSuRe algorithm, described in Algorithm 3, requires even more operations in each of its iterations, as given in Algorithm 2. In the first step, one has the RoSuRe algorithm with the associated $\mathcal{O}(PN_r^2)$ cost. The second and third mcRoSuRe steps, however, perform a distinct optimization as given in Algorithm 2, which deals

TABLE II
COMPUTATIONAL COMPLEXITY PER ITERATION OF THE EVALUATED METHODS (IN NUMBER OF MULTIPLICATIONS)

| Step | RoSuRe | mcRoSuRe | mcRoSuRe-TA | mcRoSuRe-A |
|---|---|---|---|---|
| 1 | $\mathcal{O}(PN_r^2)$ | $\mathcal{O}(PN_r^2)$ | $\mathcal{O}(PN_r^2)$ | $\mathcal{O}(PN_r^{\mathrm{ds}}N_t)$ |
| 2 | - | $\mathcal{O}\left(PN_r^2\left(1+\frac{N_t}{N_r}\right)\right)$ | $\mathcal{O}(PN_r'^2)$ | $\mathcal{O}(PN_r'^2)$ |
| 3 | - | $\mathcal{O}\left(PN_r^2\left(1+\frac{N_t}{N_r}\right)\right)$ | $\mathcal{O}\left(PN_r'^2\left(1+\frac{N_t}{N_r'}\right)\right)$ | $\mathcal{O}\left(PN_r'^2\left(1+\frac{N_t}{N_r'}\right)\right)$ |
| 4 | - | - | $\mathcal{O}\left(PN_r'^2\left(1+\frac{N_t}{N_r'}\right)\right)$ | $\mathcal{O}\left(PN_r'^2\left(1+\frac{N_t}{N_r'}\right)\right)$ |

with $P \times N_r$, $P \times N_t$, and $N_t \times N_r$ matrices. With that in mind the method needs in each iteration

$$A(N_r, N_t, P) = N_r N_t + 8PN_r \tag{23}$$

additions and

$$M(N_r, N_t, P) = 3PN_r N_t + 3N_r N_t + 3PN_t + 7 \tag{24}$$

multiplications, which results in an overall cost of $\mathcal{O}(PN_r^2 + PN_r N_t)$.

The mcRoSuRe-A algorithm, introduced in Subsection IV-A and described in Algorithm 5, creates an additional optimization step as summarized in Algorithm 4. Its first step considers an optimization on a downsampled reference video sequence containing $N_r^{\mathrm{ds}} \ll N_r$ frames. Therefore the actual number of arithmetical operations for each iteration in this step is

$$A(N_r^{\mathrm{ds}}, N_t, P) = N_r^{\mathrm{ds}} N_t + 8PN_r^{\mathrm{ds}} \tag{25}$$

additions and

$$M(N_r^{\mathrm{ds}}, N_t, P) = 3PN_r^{\mathrm{ds}} N_t + 3N_r^{\mathrm{ds}} N_t + 3PN_t + 7 \tag{26}$$

multiplications, which is dominated by the $\mathcal{O}(PN_r^{\mathrm{ds}} N_t)$ term. After this step a new reference sequence is created with only $N_r' \ll N_r$ frames, corresponding to the original reference video excerpt used to reconstruct the target frames. The following steps use the same optimization described in Algorithm 2 but using $P \times N_r'$, $P \times N_t$, and $N_t \times N_r'$ matrices, requiring in the second step

$$A(N_r', P) = 2N_r'^2 + 5PN_r' \tag{27}$$

additions and

$$M(N_r', P) = 4PN_r'^2 + 2N_r'^2 + 3PN_r' + 7 \tag{28}$$

multiplications for each iteration and in the subsequent steps

$$A(N_r', P) = N_r' N_t + 8PN_r' \tag{29}$$

additions and

$$M(N_r', N_t, P) = 3PN_r' N_t + 3N_r' N_t + 3PN_t + 7 \tag{30}$$

multiplications for each iteration.

These operations lead to an overall cost of the order $\mathcal{O}(PN_r'^2)$ for the second step and $\mathcal{O}(PN_r'^2 + PN_r' N_t)$ for the third and fourth ones, which once again is much smaller than $\mathcal{O}(PN_r^2)$ and $\mathcal{O}(PN_r^2 + PN_r N_t)$ respectively, as $N_r' \ll N_r$.

A summary of the final computational complexities of the algorithms analyzed is given in Table II. From the above analysis, one can infer that mcRoSuRe-TA and mcRoSuRe-A reduce drastically the resulting overall complexity when compared with mcRoSuRe, as verified quantitatively in Section V.

To provide numerical information on these computational complexities, we show here the figures associated with an example from Section V-B. In this experimental scenario, based on a real-world application, $R = 320$, $C = 180$, yielding $P = 57600$, $N_r = 5000$ and $N_t = 200$. Using a typical value for the downsampling value one will have $N_r^{\mathrm{ds}} = 500$. Provided that the camera does not stop during the translational movement (common case in real applications) $N_r' = 210$ (the size of $N_t$ plus a guard interval). For a list of the variables refer to Table I.

With these values the mcRoSuRe method would have the following numbers of multiplications per iteration

- First step: $9.14 \cdot 10^8$ multiplications
- Second step: $1.73 \cdot 10^{11}$ multiplications
- Third step: $1.73 \cdot 10^{11}$ multiplications

while mcRoSuRe-A would have

- First step: $1.73 \cdot 10^{10}$ multiplications
- Second step: $1.02 \cdot 10^{10}$ multiplications
- Third step: $7.29 \cdot 10^9$ multiplications
- Fourth step: $7.29 \cdot 10^9$ multiplications

The gains in computation complexity in every step by using the proposed algorithm can be inferred from this example.

## V. EXPERIMENTAL RESULTS

### A. Video Database for Abandoned Object Detection

To test the performance of the proposed algorithms in a real-world challenging scenario the Video Database for Abandoned Object Detection (VDAO) database (described in [45] and available for download at [46]) was used. This database contains over 8 hours of videos recorded in visually cluttered complex environments of industrial plants. The database videos contain several challenges as illumination variation, occlusion of objects, and camera jitter. The abandoned objects are everyday static objects placed in the industrial scenario. All videos feature reference and target sequences with manually marked ground-truth labels. To the authors' best knowledge, the VDAO database is the only publicly available one exclusively designed for the detection of abandoned objects [47].

The VDAO database was recorded using a camera mounted on top of a moving robotic platform that follows a linear path of about 6 m on a hanging rail at a height of approximately 2.5m. The camera is pointed at a cluttered environment comprised of several pipes and valves depicting a scene of interest inside an industrial facility. The database videos are

TABLE III
TIME (IN SECONDS) USED BY EACH STEP OF THE MCROSURE, MCROSURE-TA, AND MCROSURE-A METHODS WHEN ANALYZING THE VDAO DATABASE WITH DIFFERENT REFERENCE/TARGET VIDEO LENGTHS

| Short Videos - $N_r = 200$ and $N_t = 100$ frames | | | |
|---|---|---|---|
| Step | mcRoSuRe | mcRoSuRe-TA | mcRoSuRe-A |
| 1 | 44.09 | 17.45 | **9.77** |
| 2 | 16.70 | **12.66** | 13.57 |
| 3 | 16.15 | **12.79** | 12.78 |
| 4 | - | 14.29 | **14.23** |
| **Total** | 76.94 | 57.19 | **50.34** |
| Medium Videos - $N_r = 1000$ and $N_t = 200$ frames | | | |
| Step | mcRoSuRe | mcRoSuRe-TA | mcRoSuRe-A |
| 1 | 764.65 | 95.69 | **29.05** |
| 2 | 97.01 | 58.85 | **58.56** |
| 3 | 86.79 | 36.75 | **35.83** |
| 4 | - | 38.66 | **36.35** |
| **Total** | 948.46 | 229.97 | **159.80** |
| Long Videos - $N_r = 5000$ and $N_t = 200$ frames | | | |
| Step | mcRoSuRe | mcRoSuRe-TA | mcRoSuRe-A |
| 1 | 27333.18 | 537.20 | **66.06** |
| 2 | 529.32 | **122.31** | 124.43 |
| 3 | 477.92 | 50.55 | **49.10** |
| 4 | - | **48.25** | 52.14 |
| **Total** | 28340.42 | 758.31 | **291.73** |

separated in two groups: single- and multi-object videos. The single-object videos have only one abandoned object placed along the camera path, whereas the multi-object videos have at least two abandoned objects present in every frame of the video. All the videos contain several passes of the camera in the rail aiming to the same region.

Although the VDAO videos are in full-color and with $1280 \times 720$-pixel resolution, for the proposed experiments all videos were converted to grayscale and downsampled to a $320 \times 180$ resolution.

### B. Experimental Assessment of the Proposed Algorithms

In a first experiment we compare the three versions of the mcRoSuRe algorithm: the original one summarized in Algorithm 3 (mcRoSuRe) [14], the one in Algorithm 5 (mcRoSuRe-TA) [44], and the accelerated one proposed here (mcRoSuRe-A) which uses a 10:1 decimated version of the reference video in this first step of the algorithm.

For comparison purposes, we evaluate the performances of these three versions when matrix $\mathbf{X}_r$ is composed by $N_r = \{5000, 1000, 200\}$ frames of a given VDAO reference video and $\mathbf{X}_r$ is comprised of $N_t = \{200, 200, 100\}$ frame excerpts, respectively, from each of the 59 single-object VDAO target videos. As for parameter initialization, we used: $\lambda = 1$, $\rho = 1.5$, $\eta_1 = 3$, $\eta_2 = 1.1\sigma_1(\mathbf{X}_r)$, and $\mu_0 = 1.25/\sigma_1(\mathbf{X}_r)$, where $\sigma_1(\mathbf{X}_r)$ is the largest singular value of input matrix $\mathbf{X_r}$.

Table III shows the time (in seconds) taken by each algorithm step when analyzing all videos in an Intel i7-3630QM with 2.4GHz and 16GB of RAM, running MATLAB © 2012b. From this table, it is noticeable how the proposed modifications accelerate the algorithm, particularly in the first step which is the dominant one in the original mcRoSuRe version. Comparing the total running time of each algorithm, one notices how the proposed mcRoSuRe-A (using 10:1 downsampling ratio) outperformed the other two, specially for longer video sequences where the acceleration factor becomes 2.6 with respect to the mcRoSuRe-TA and 100 with respect to mcRoSuRe.

It must be emphasized that this speed improvement occurs without hindering the system's detection capability. In fact, when one compares the outputs of both mcRoSuRe and mcRoSuRe-A methods, one readily observes that both methods have very similar (if not exactly equal) results, as depicted in Fig. 4. Similar results for the mcRoSuRe-TA method can be found in [44].

### C. Abandoned Object Detection Algorithms Using Moving Camera

The performance of the proposed mcRoSuRe-A algorithm has been assessed by comparing it with those of some of state-of-the-art methods, such as the detection of abandoned objects with a moving camera (DAOMC) [29], the moving-camera background subtraction (MCBS) [34], and the spatio-temporal composition for moving-camera detection (STC-mc) [36]. To this end we used the annotated videos from the VDAO database. As the algorithms of [29], [34], and [36] could not be executed in a reasonable amount of time for the complete VDAO videos, only short-length 200-frame videos were employed.

The selected 200-frames video excerpts used in the experiments described in this paper are publicly available at [48]. The results of all experiments carried out with the competing methods can also be found at [48].

For comparison purposes, the reference-target video synchronization was performed manually for the DAOMC algorithm. In our implementation of the DAOMC, an NCC window small enough to detect all objects in the database was used to allow a fair comparison. For the MCBS algorithm, since the application of the method changed from a railway surveillance problem to a more general scenario, the post-processing steps that find the railway tracks were removed from the original algorithm. For the STC-mc the original author's implementation of the algorithm was used. In addition, the results presented for the MCBS algorithm were obtained after the application of the optimized parameter configuration for the two similarity metrics used in the original paper [34], namely: the normalized vector distance (NVD) [49] and the radial reach filter (RRF) [50].

To obtain quantitative results for the mcRoSuRe-A algorithm the output matrix $\mathbf{E}_e$ was post-processed with simple open and close morphological operations with 1 to 5 pixel-wide structuring elements. Also, simple binary thresholding was applied to obtain the final detection mask.

The performance for all methods was initially quantified with the following metrics: (i) True positive (TP) detection rate, where a TP occurs when the detection blob has at least one coincident pixel with the abandoned-object ground-truth bounding box; (ii) False positive (FP) detection rate, where an

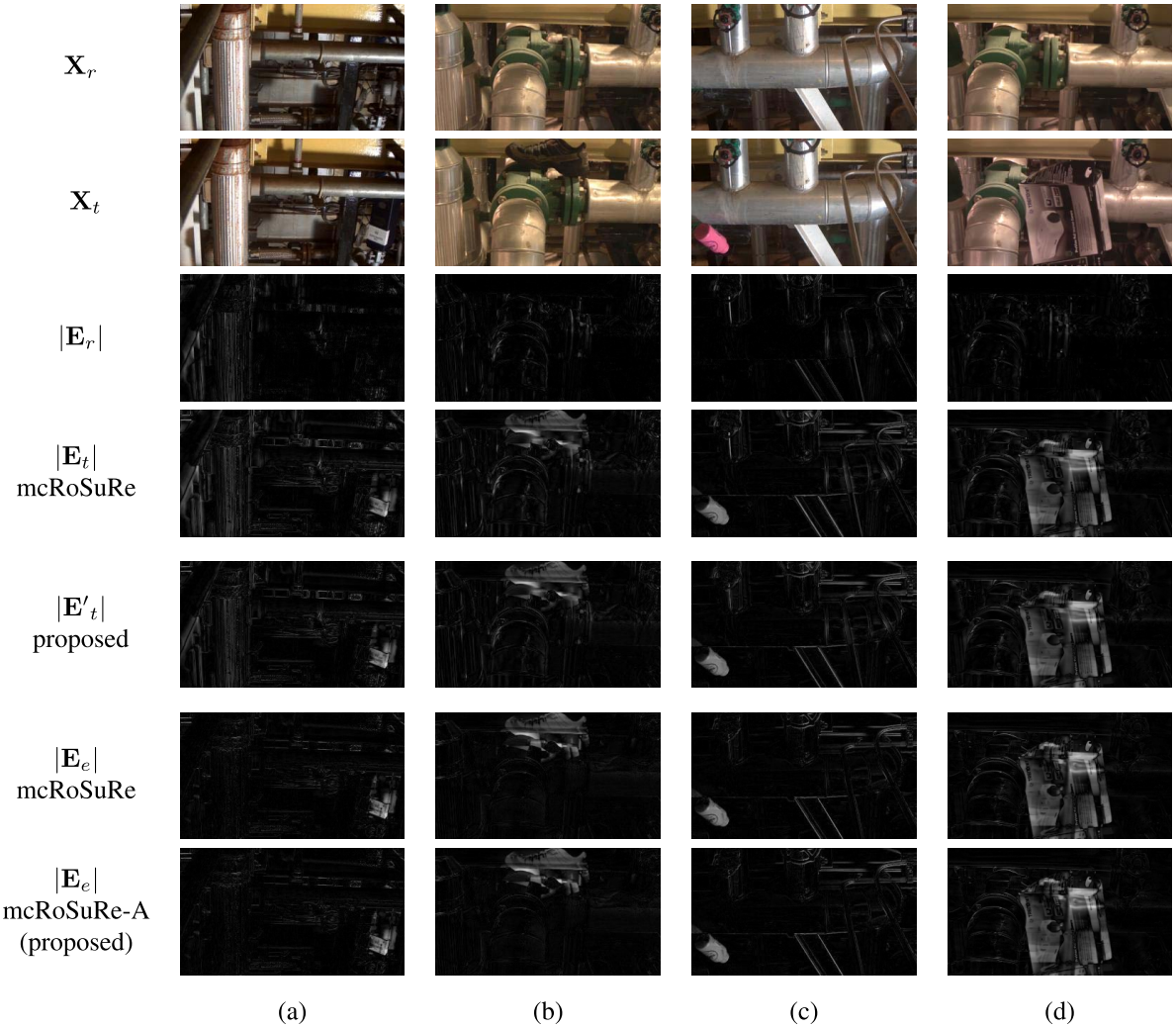


Fig. 4. Comparative results for the mcRoSuRe and mcRoSuRe-A algorithms (single frames of matrices $X_r$, $X_t$, $E_r$, $E_t$, and $E$ of both methods) for 4 different abandoned-object videos from VDAO [45] database: (a) blue box; (b) shoe (c) pink bottle; (d) camera box. The similar detection performance of both methods is clear from these experiments.

FP arises when the detection blob has all pixels outside the ground-truth bounding box; (iii) False negative (FN) detection rate, where an FN occurs when the ground-truth bounding box has no detected pixels inside it; (iv) True negative (TN) detection rate, where a TN is associated to a frame with no bounding box and no detected pixels. In addition, we consider the DIS parameter defined as

$$\text{DIS} = \sqrt{(1 - \text{TP})^2 + \text{FP}^2}, \tag{31}$$

which can be interpreted as the minimum distance of all operating points to the point of ideal behaviour (TP $= 1$ and FP $= 0$) in the TP$\times$FP plane. The use of this metric allows direct comparison with the results in [36].

In a first experiment, the same seven video excerpts of [36] were considered. Since those videos contain only frames with objects, only the TP and FP measurements are shown in Table IV along with the distance parameter.

It is clear from the results in Table IV that for those limited scenarios considered in [36] the mcRoSuRe-A method is either equivalent or superior to the other algorithms for all considered metrics. The average mcRoSuRe-A TP of 0.99 shows that in almost all the cases the algorithm is able to detect the presence of anomalies, with a somewhat low FP detection rate of 0.20. The lowest DIS value of 0.20 indicates that the mcRoSuRe-A algorithm achieves the best balance between the TP and FP detections for this problem among all the competing algorithms.

In a more extensive analysis, we considered the algorithm average performances for all 59 single-object VDAO videos, as given in Table V. In these videos there are both frames with and without objects.

By analyzing the results presented in Table V one notices that the mcRoSuRe-A method is consistently superior to the other three competing methods in every metric considered. The average mcRoSuRe-A TP detection rate is the only one above 0.90, while yielding the lowest average FP detection rate. Unlike the competing algorithms mcRoSuRe-A provides over 0.60 of TN detections. In the case of the VDAO database this is most challenging metric as even small changes in

TABLE IV

DETECTION COMPARISON OF PROPOSED MCROSURE-A METHOD WITH THAT OF STC-MC, DAOMC, AND MCBS METHODS FOR THE SAME SEVEN VIDEOS EXTRACTS EMPLOYED IN [36]

| Object | STC-mc | | | DAOMC | | | MCBS | | | mcRoSuRe-A | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 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 | 0.96 | 0.17 | 0.17 |
| Towel | 0.92 | 0.01 | 0.08 | 1.00 | 0.10 | 0.10 | **1.00** | **0.07** | **0.07** | 0.99 | 0.47 | 0.47 |
| 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 | 0.75 | 0.75 |
| **Average** | 0.78 | 0.19 | 0.59 | 1.00 | 0.32 | 0.32 | 1.00 | 0.47 | 0.47 | **0.99** | **0.20** | **0.20** |

TABLE V

AVERAGE DETECTION COMPARISON OF PROPOSED MCROSURE-A METHOD WITH THAT OF STC-MC, DAOMC, AND MCBS METHODS FOR ALL 59 SINGLE-OBJECT VIDEOS OF THE VDAO DATABASE

| Method | TP | FP | TN | FN |
|---|---|---|---|---|
| STC-mc | 0.18 | 0.38 | 0.59 | 0.82 |
| DAOMC | 0.83 | 0.43 | 0.54 | 0.17 |
| MCBS | 0.89 | 0.84 | 0.02 | 0.11 |
| mcRoSuRe-A | **0.91** | **0.33** | **0.63** | **0.09** |

TABLE VI

AVERAGE DETECTION COMPARISON OF PROPOSED MCROSURE-A METHOD WITH THAT OF STC-MC, DAOMC, AND MCBS METHODS FOR ALL 59 SINGLE-OBJECT VIDEOS OF THE VDAO DATABASE USING FRAME-LEVEL METRICS

| Method | $TP_{fl}$ | $FP_{fl}$ | $TN_{fl}$ | $FN_{fl}$ |
|---|---|---|---|---|
| STC-mc | 0.48 | 0.41 | 0.59 | 0.52 |
| DAOMC | 0.89 | 0.46 | 0.54 | 0.11 |
| MCBS | **0.99** | 0.98 | 0.02 | **0.01** |
| mcRoSuRe-A | 0.95 | **0.37** | **0.63** | 0.05 |

TABLE VII

AVERAGE DETECTION COMPARISON OF PROPOSED MCROSURE-A METHOD WITH THAT OF STC-MC, DAOMC, AND MCBS METHODS FOR THE 9 MULTI-OBJECT VIDEOS OF THE VDAO DATABASE

| Method | TP | FP | DIS |
|---|---|---|---|
| STC-mc | 0.67 | 0.74 | 0.81 |
| DAOMC | **1.00** | 0.68 | 0.68 |
| MCBS | **1.00** | 0.59 | 0.59 |
| mcRoSuRe-A | 0.96 | **0.25** | **0.25** |

illumination and camera position can yield false detections. Finally the mcRoSuRe-A is the only one among the tested methods to have less than 0.10 average FN, providing the least amount of undetected anomalies.

In this experiment we used the parameter values tuned for the initial seven video experiment shown in Table IV for all the compared methods. Since the videos in this experiment present more challenging features (as objects being occluded) and a larger variation in objects shapes and illuminations, not all methods kept their good results. In contrast with most of the competing methods mcRoSuRe-A have shown to be robust to the challenges presented in this database having shown the least decrease in the performance when compared with the initial test results.

If one is not concerned with the identification of the anomaly position inside a given frame, but wants only to determine whether a frame presents an anomaly, a more relaxed version of the detection metrics can be used. By considering only a frame-level detection analysis, one may define a $TP_{fl}$ (or $FP_{fl}$) by the presence of any detection blob in an anomalous (non-anomalous) frame and an $FN_{fl}$ (or $TN_{fl}$) by the absence of a detection blob in an anomalous (non-anomalous) frame. Average results for these frame-level metrics for all four detection algorithms and for all 59 single-object videos from the VDAO database are shown in Table VI.

Table VI leads to similar conclusions as Table V. Since here the localization of the anomaly inside the frame is no longer an issue, then slightly misplaced detection blobs now count as a correct detection thus making the small objects more frequently detected by all methods, improving, for instance, the $TP_{fl}$ mcRoSuRe-A measurement to 0.95. Although the $TP_{fl}$ results for the MCBS method are superior to the ones of mcRoSuRe-A, it yields also yields 0.99 of $FP_{fl}$ detection, showing it is unreliable for this type of application. On the other hand, the $FP_{fl}$ also increased for all methods, as now only the frames where there are no anomalies count for this verification, thus making every error more relevant on the statistics.

Another test was performed using the multi-object videos from the VDAO database. Those videos are much more challenging than the single-object videos, as in this case, there are very small objects that can be hard to detect. Also, the contrast of the videos is not as good as that of the single-object videos. The results of these experiments are summarized in Table VII.

Since in the multi-object videos each frame has at least two objects (as explained in Section V-A), there are no TN frames. Thus, as a result, similarly to what happened with the 7-video tests, only the TP, FP, and DIS results are displayed. Unfortunately, by using the metrics that were chosen for the other experiments it is not possible to take into account the number of objects that were correctly detected in a frame with more than one object.

When analysing the results from this experiment, it is clear again that, in this more challenging scenario, mcRoSuRe-A

TABLE VIII
TIME (IN SECONDS) USED BY ALGORITHMS STC-MC, DAOMC, MCBS, AND MCROSURE-A METHODS WHEN ANALYZING SEVEN VIDEOS FROM THE VDAO DATABASE

| | STC-mc | DAOMC | MCBS | mcRoSuRe-A |
|---|---|---|---|---|
| Dark blue box 1 | 433 | 265 | 50924 | 52 |
| Towel | 345 | 280 | 50403 | 38 |
| Shoe | 542 | 293 | 50427 | 38 |
| Pink bottle | 415 | 280 | 50170 | 38 |
| Camera box | 448 | 299 | 50238 | 45 |
| Dark blue box 2 | 221 | 289 | 51740 | 38 |
| White jar | 248 | 282 | 49901 | 36 |
| **Average** | 379 | 284 | 50543 | 41 |

presents more reliable results than the other compared methods. Although DAOMC and MCBS have better TP results those two methods present much higher FP results as well, as can be seen by inspecting the DIS measurement in the last column of Table VII.

Finally, the time performance of all the competing algorithms was compared using a computer with Intel i7-4790K with 4.0GHz and 32GB of RAM, running MATLAB ©2015a. Table VIII presents the total time taken by each algorithm to run the same seven videos considered in Table IV. From these results, one can easily notice how the mcRoSuRe-A method is the fastest one, being able to run at least seven times faster than the other methods in this test.

## VI. CONCLUSION

This paper presents a family of algorithms that use sparse representations for detecting anomalies in video sequences obtained from slow moving cameras. The proposed techniques project the acquired data from a reference (anomaly-free) video onto a union of subspaces, and select a small number of those subspaces that contain most of the information needed to reconstruct the target (possibly anomalous) video.

The present work has shown the efficiency of the mcRoSuRe-A method demonstrating that it is able to cope with challenging scenarios in much less processing time than the other methods in mcRoSuRe family, while attaining qualitatively similar results. Depending on the size of the videos, the method was shown to be able to run up to 2.6 times faster than mcRoSuRe-TA [44] and 100 times faster than the original mcRoSuRe [14] algorithm, placing it among the fastest methods for anomaly detection in moving-camera videos.

Extensive experiments were conducted comparing the mcRoSuRe-A detection performance with alternative state-of-the-art approaches using the challenging VDAO database. The algorithm was shown to perform well in this database attaining the best average performance in all tests, reaching an average rate of 0.91 of true positive detections and around 0.33 of false positive detection, having the best compromise among the tested methods.

## ACKNOWLEDGMENT

The authors would like to thank Xiao Bian for the availability in answering questions for the implementation of the algorithms.

## REFERENCES

[1] T. M. Research, "Video surveillance and VSaaS market—Global industry analysis, size, share, growth, trends and forecast 2016–2024," TMS Anal., Albany, NY, USA, Tech. Rep., Apr. 2016.
[2] Y. Tian, R. S. Feris, H. Liu, A. Hampapur, and M.-T. Sun, "Robust detection of abandoned and removed objects in complex surveillance videos," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 5, pp. 565–576, Sep. 2011.
[3] E. Hossain and G. Chetty, "Person identification in surveillance video using gait biometric cues," in *Proc. 9th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Sichuan, China, May 2012, pp. 1877–1881.
[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Comput. Res. Repository*, vol. abs/1512.03385, pp. 1–12, Dec. 2015. [Online]. Available: http://arxiv.org/abs/1512.03385
[5] O. M. Sincan, V. B. Ajabshir, H. Y. Keles, and S. Tosun, "Moving object detection by a mounted moving camera," in *Proc. IEEE Int. Conf. Comput. Tool*, Salamanca, Spain, Nov. 2015, pp. 1–6.
[6] P. Singh, B. B. V. L. Deepak, T. Sethi, and M. D. P. Murthy, "Real-time object detection and tracking using color feature and motion," in *Proc. IEEE Int. Conf. Commun. Signal Process.*, Melmaruvathur, India, Apr. 2015, pp. 1236–1241.
[7] A. Taneja, L. Ballan, and M. Pollefeys, "Geometric change detection in urban environments using images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 11, pp. 2193–2206, Nov. 2015.
[8] M. Bengel, K. Pfeiffer, B. Graf, A. Bubeck, and A. Verl, "Mobile robots for offshore inspection and manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robost Syst.*, St. Louis, MO, USA, Oct. 2009, pp. 3317–3322.
[9] E. Kyrkjebø, P. Liljebäck, and A. A. Transeth, "A robotic concept for remote inspection and maintenance on oil platforms," in *Proc. Int. Conf. Ocean, Offshore Arctic Eng.*, Honolulu, HI, USA, Jun. 2009, pp. 667–674.
[10] JPT Staff, "Sensabot: A safe and cost-effective inspection solution," *J. Petroleum Technol.*, vol. 64, no. 10, pp. 32–34, 2012.
[11] M. Galassi *et al.*, "DORIS—A mobile robot for inspection and monitoring of offshore facilities," in *Proc. Congr. Brasileiro Autom.*, Belo Horizonte, Brazil, Sep. 2014, pp. 3174–3181.
[12] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, May 2011.
[13] X. Bian and H. Krim, "Bi-sparsity pursuit for robust subspace recovery," in *Proc. IEEE Int. Conf. Image Process.*, Quebec City, QC, Canada, Sep. 2015, pp. 3535–3539.
[14] E. Jardim, X. Bian, E. A. B. da Silva, S. L. Netto, and H. Krim, "On the detection of abandoned objects with a moving camera using robust subspace recovery and sparse representation," in *Proc. IEEE Int. Conf. Acoust., Int. Conf. Speech Signal Process.*, Brisbane, QLD, Australia, Apr. 2015, pp. 1295–1299.
[15] R. Miezianko and D. Pokrajac, "Detecting and recognizing abandoned objects in crowded environments," in *Proc. Int. Conf. Comput. Vis. Syst.*, Santorini, Greece, May 2008, pp. 241–250.
[16] C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Robust principal component analysis for background subtraction: Systematic evaluation and comparative analysis," in *Principal Component Analysis*. Rijeka, Croatia: InTech, 2012, ch. 12.
[17] L. Chang, H. Zhao, S. Zhai, Y. Ma, and H. Liu, "Robust abandoned object detection and analysis based on online learning," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Shenzhen, China, Dec. 2013, pp. 940–945.
[18] Wahyono, A. Filonenko, and K.-H. Jo, "Detecting abandoned objects in crowded scenes of surveillance videos using adaptive dual background model," in *Proc. Int. Carnahan Conf. Secur. Technol.*, Lexington, KY, USA, Jun. 2015, pp. 224–227.
[19] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *Proc. Int. Conf. Syst., Signals Image Process.*, Bratislava, Slovakia, May 2016, pp. 1–4.
[20] J. He, D. Zhang, L. Balzano, and T. Tao, "Iterative Grassmannian optimization for robust image alignment," *Image Vis. Comput.*, vol. 32, no. 10, pp. 800–813, Oct. 2014.
[21] J. Xu, V. K. Ithapu, L. Mukherjee, J. M. Rehg, and V. Singh, "GOSUS: Grassmannian online subspace updates with structured-sparsity," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3376–3383.
[22] W. Song, J. Zhu, Y. Li, and C. Chen, "Image alignment by online robust PCA via stochastic gradient descent," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 7, pp. 1241–1250, Jul. 2016.
[23] P. Rodríguez and B. Wohlberg, "Translational and rotational jitter invariant incremental principal component pursuit for video background modeling," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, pp. 537–541.

[24] S. Javed, S. K. Jung, A. Mahmood, and T. Bouwmans, "Motion-aware graph regularized RPCA for background modeling of complex scenes," in *Proc. Int. Conf. Pattern Recognit.*, Dec. 2016, pp. 120–125.
[25] S. Javed, A. Mahmood, T. Bouwmans, and S. K. Jung, "Spatiotemporal low-rank modeling for complex scene background initialization," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.
[26] X. Zhou, C. Yang, H. Zhao, and W. Yu, "Low-rank modeling and its applications in image analysis," *CoRR*, vol. abs/1401.3409, pp. 1–35, Oct. 2014.
[27] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah, "Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset," *Comput. Sci. Rev.*, vol. 23, pp. 1–71, Feb. 2017.
[28] A. Sobral, T. Bouwmans, and E.-H. Zahzah, "LRSLibrary: Low-rank and sparse tools for background modeling and subtraction in videos," in *Handbook of Robust Low-Rank and Sparse Matrix Decomposition Applications in Image and Video Processing*. Boca Raton, FL, USA: CRC Press, 2014.
[29] H. Kong, J.-Y. Audibert, and J. Ponce, "Detecting abandoned objects with a moving camera," *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 2201–2210, Aug. 2010.
[30] F. Diego, D. Ponsa, J. Serrat, and A. M. López, "Video alignment for change detection," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1858–1869, Jul. 2011.
[31] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[32] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[33] G. Carvalho, "Automatic detection of abandoned objects with a moving camera using multiscale video analysis," Ph.D. dissertation, COPPE-Univ. Fed. Rio de Janeiro, Rio de Janeiro, Brazil, 2015.
[34] H. Mukojima *et al.*, "Moving camera background-subtraction for obstacle detection on railway tracks," in *Proc. IEEE Int. Conf. Image Process.*, Phoenix, AZ, USA, Sep. 2016, pp. 3967–3971.
[35] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Moving camera background-subtraction for obstacle detection on railway tracks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 1385–1392.
[36] M. T. Nakahata, L. A. Thomaz, A. F. da Silva, E. A. B. da Silva, and S. L. Netto, "Anomaly detection with a moving camera using spatio-temporal codebooks," in *Multidimensional Systems and Signal Processing*. New York, NY, USA: Springer, Mar. 2017, pp. 1–30.
[37] L. A. Thomaz, A. F. da Silva, E. A. B. da Silva, S. L. Netto, X. Bian, and H. Krim, "Abandoned object detection using operator-space pursuit," in *Proc. IEEE Int. Conf. Image Process.*, Quebec City, QC, Canada, Sep. 2015, pp. 1980–1984.
[38] X. Bian and H. Krim, "Optimal operator space pursuit: A framework for video sequence data analysis," in *Computer Vision* (Lecture Notes in Computer Science), vol. 7725. Berlin, Germany: Springer, 2013, pp. 760–769.
[39] X. Bian and H. Krim, "Robust subspace recovery via dual sparsity pursuit," *Comput. Res. Repository*, vol. abs/1403.8067 pp. 1–17, Mar. 2014. [Online]. Available: http://arxiv.org/abs/1403.8067
[40] W.-Y. Yan, "On principal subspace analysis," *J. Franklin Inst.*, vol. 335, no. 4, pp. 707–718, 1998.
[41] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer, 2002.
[42] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 1990.
[43] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, Jan. 2014.
[44] L. A. Thomaz, A. F. da Silva, E. A. B. da Silva, S. L. Netto, and H. Krim, "Detection of abandoned objects using robust subspace recovery with intrinsic video alignment," in *Proc. IEEE Int. Conf. Circuits Syst.*, Baltimore, MD, USA, May 2017, pp. 1–4.
[45] A. F. D. Silva, L. A. Thomaz, G. Carvalho, M. T. Nakahata, and E. Jardim, "An annotated video database for abandoned-object detection in a cluttered environment," in *Proc. Int. Telecommun. Symp. (ITS)*, Sao Paulo, Brazil, Aug. 2014, pp. 1–5.
[46] (2014). *VDAO—Video Database of Abandoned Objects in a Cluttered Industrial Environment*. [Online]. Available: http://www.smt.ufrj.br/~tvdigital/database/objects
[47] C. Cuevas, R. Martínez, and N. García, "Detection of stationary foreground objects: A survey," *Comput. Vis. Image Understand.*, vol. 152, pp. 41–57, Nov. 2016.
[48] (2017). *200-Frame Excerpts Form VDAO Database*. [Online]. Available: http://www02.smt.ufrj.br/~tvdigital/database/research/
[49] T. Matsuyama, T. Ohya, and H. Habe, "Background subtraction for non-stationary scene," in *Proc. Asian Conf. Comput. Vis.*, Taipei, Taiwan, Aug. 2000, pp. 662–667.
[50] Y. Satoh, H. Tanahashi, C. Wang, S. Kaneko, Y. Niwa, and K. Yamamoto, "Robust event detection by radial reach filter (RRF)," in *Proc. Int. Conf. Pattern Recognit.*, vol. 2. Quebec City, QC, Canada, Aug. 2002, pp. 623–626.

**Lucas A. Thomaz** (S'14) 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.

**Eric Jardim** was born in Salvador, Brazil. He received the B.Sc. degree in computer science from the Universidade Federal da Bahia in 2003, and the M.Sc. degree in mathematics from the Instituto Nacional de Matemática Pura e Aplicada in 2010. He is currently pursuing the D.Sc. degree in electrical engineering with the Universidade Federal do Rio de Janeiro. Since 2003, he was with Petróleo Brasileiro S.A. (PETROBRAS) involved in scientific computing, visualization, and robotics. His research interests are in computer vision, machine learning, and non-photorealistic rendering.

**Allan F. da Silva** (S'14) was born in Brazil in 1990. He received the B.E. degree in electronic and computer engineering and the M.Sc. degree in electrical engineering 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** (M'95–SM'05) 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, U.K., 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 is the Deputy Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I. He has been a Distinguished Lecturer of the IEEE Circuits and Systems Society.

**Sergio L. Netto** (SM'04) 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.

**Hamid Krim** (SM'99–F'08) received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Washington, Seattle, WA, USA, and the Ph.D. degree in electrical and computer engineering from Northeastern University, Boston, MA, USA. He was a member of Technical Staff at AT&T Bell Labs, where he has conducted research and development in the areas of telephony and digital communication systems/subsystems. Following an NSF Post-Doctoral Fellowship at the Foreign Centers of Excellence, LSS/University of Orsay, Paris, France, he joined the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA, as a Research Scientist, where he was performing and supervising research.

He is currently a Professor of electrical engineering with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, USA, leading the Vision, Information, and Statistical Signal Theories and Applications Group. His research interests include statistical signal and image analysis and mathematical modeling with a keen emphasis on applied problems in classification and recognition using geometric and topological tools. He has served on the SP Society Editorial Board and on TCs. He is the SP Distinguished Lecturer from 2015 to 2016.