Domain-Transformable Sparse Representation for Anomaly Detection in Moving-Camera Videos

Eric Jardim, Lucas A. Thomaz, *Member, IEEE,* Eduardo A. B. da Silva, *Senior Member, IEEE,* and Sergio L. Netto, *Senior Member, IEEE.*

***Abstract*—This paper presents a special matrix factorization based on sparse representation that detects anomalies in video sequences generated with moving cameras. Such representation is made by associating the frames of the target video, that is a sequence to be tested for the presence of anomalies, with the frames of an anomaly-free reference video, which is a previously validated sequence. This factorization is done by a sparse coefficient matrix, and any target-video anomaly is encapsulated into a residue term. In order to cope with camera trepidations, domain-transformations are incorporated into the sparse representation process. Approximations of the transformed-domain optimization problem are introduced to turn it into a feasible iterative process. Results obtained from a comprehensive video database acquired with moving cameras on a visually cluttered environment indicate that the proposed algorithm provides a better geometric registration between reference and target videos, greatly improving the overall performance of the anomaly-detection system.**

***Index Terms*—Video anomaly detection, sparse representation, matrix factorization, object detection, change detection, moving camera, l_1-optimization.**

I. Introduction

ANOMALY detection in images and video sequences is a classical research problem in computer vision and related areas, which has direct applications in many tasks, ranging from domestic security and medical diagnosis to industrial and military activities [1], [2]. The increasing number of applications and the necessity of precise results are raising the demand for alternative, less human-dependant solutions. Apart from being a regularly known problem, automatic anomaly detection still remains a difficult and challenging topic due to several complex issues such as camera pose, illumination, shadows, occlusions, weather conditions, camera jitter, and so on [3].

In several surveillance tasks, additional cameras should be employed to deal with the problem of multiple occlusions. Some activities, especially in cluttered environments like industrial plants and offshore oil platforms, usually require multiple viewpoints for proper inspection [4], [5]. Such a need is even greater in hazardous environments and when there are places that are difficult to access [6], [7], [8]. Increasing the number of cameras also increases the amount of video to be analyzed what can become unpractical in large facilities.

An interesting approach to deal with this issue is to monitor several viewpoints with a single moving camera. In practice, a conventional camera can be mounted on top of a moving platform (e.g. a car, robot, or drone) that takes the camera to the desired positions along a predefined trajectory. While this approach can significantly reduce the necessary number of cameras, it also enables the selection of specific points-of-view to be monitored at the same time that it allows the automation of repetitive inspections. These factors, allied to the widespread use of portable cameras, are spurring the interest in problems of surveillance and background/foreground separation using moving cameras [6], [9], [10], [11], [12].

The work presented in this paper addresses the problem of detecting changes in video sequences captured by this kind of camera arrangements. The proposed method decomposes a possibly anomalous target video into a sparse combination of the frames from an anomaly-free reference video plus a sparse residual that corresponds to the possible anomalies of interest. The basic assumption for the algorithm to work efficiently is that the camera's poses and trajectories during the target and reference video acquisitions are similar, in such a way that the information in the frames of the target video is mostly contained in the frames from the reference video. Under these circumstances, the decomposition of the target frames as an sparse combination of the reference frames can be achieved by linear convex optimization. Unfortunately, in real world scenarios, the camera trembles as it moves, and thus its pose and trajectory during recording of the reference video may present variations relative to the recording of the target video. A way to cope with this issue is to add to the optimization process an additional non-linear domain-transformation term, that will require an iterative linearization approximation. This domain transformation enables the method to find a better correspondence between reference and target video frames, thus yielding less false detections as result of the algorithm.

The main contributions of the present work are related to the expansion of the anomaly detection capabilities of low-rank representation methods, such as the mcRoSuRe-A [8], by incorporating domain transformations on the optimization framework. That improvement allows such methods to achieve a better correspondence between reference and target frames, thus obtaining superior anomaly-detection results. The proposed approach main innovation consists in the development of a sparse representation model that incorporates domain transformations in its iterative optimization procedure by using

Eric Jardim, Lucas A. Thomaz, Eduardo A. B. da Silva, and Sergio L. Netto are with the Electrical Engineering Program at COPPE - Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brazil. e-mail: {eric.jardim,lucas.thomaz,eduardo,sergioln}@smt.ufrj.br Lucas A. Thomaz is also with the Instituto de Telecomunicações, Portugal. Eduardo A. B. da Silva was supported in part by FAPERJ, grant no E-26/202.856/2018 and CNPq grant no. 307675/2014-9. This work was funded by FCT/MEC through national funds, under project PlenoISLA PTDC/EEI-TEL/28325/2017 and when applicable co-funded by FEDER – PT2020 partnership agreement under the project UID/EEA/50008/2019.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2940686, IEEE Transactions on Image Processing

a first-order approximation of the transformations. In such framework the proposed method performs geometric transformation only in the target frames, hence not changing the reference video (or dictionary). This improves the computational efficiency of the modeling process in comparison to other domain transformation methods [13].

This text is organized as follows: Section II reviews the state-of-the-art techniques for linear low-rank modeling of high-dimensional data by matrix decomposition with sparse representations. Section III introduces the problem of anomaly detection in videos acquired from moving cameras and details the adaptation of one of these sparse representation-based techniques for the problem at hand. Section IV then discusses how to make such a scheme robust to severe geometric misalignment between reference and target videos. The idea is to incorporate a transformation into the optimization problem associated to the sparse representation. Section V validates the proposed methodology by presenting experimental results on a comprehensive video database, comparing it to the state-of-the-art. Finally, conclusions are drawn in Section VI emphasizing the paper main contributions.

II. Literature Review

A. Sparse and low-rank decompositions

1) Principal component analysis: When dealing with high-dimensional observations, a very popular and successful approach is to fit the data with a simplified, lower-dimensional model. More precisely, data is often assumed to lie approximately on some low-rank subspace, reducing model complexity and consequently increasing model robustness and simplifying further analysis and storage space. Unfortunately, real-world data usually comes from sensors, which may suffer from noise and other types of perturbations. Thus, these sensor readings can be modeled as the superposition of a low-rank component plus some kind of undesired corruption term. Mathematically, if $X \in \mathbb{R}^{m\times n}$ represents a matrix which columns are these observed values, it can be modeled as

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

where the columns of L represents a low-rank model and E is a matrix of perturbations. In practice, the problem of finding anomalies is reduced to the decomposition of X, isolating the spurious data into the perturbation component E.

In statistics, *principal component analysis* (PCA) is perhaps one of the most commonly known tools for data analysis, and it is widely used in this kind of scenario. If $r = \text{rank}(L)$ is previously known and the entries of E are relatively small, with independent and identical Gaussian distribution, the problem can be efficiently solved with PCA by simply performing a *singular value decomposition* (SVD) of X and projecting its columns onto the subspace spanned by the r major left-singular vectors obtained in the process. Under these circumstances, the estimated subspace is optimal in the sense that it minimizes the mean squared reconstruction error of the columns of X, which allows us to rewrite the PCA as the optimization problem

$$\min_{L,E} ||E||_F \quad \text{subject to (s.t.)} \begin{cases} X = L + E \\ \text{rank}(L) \leq r \end{cases}, \tag{2}$$

where $||.||_F$ is the Frobenius norm of a given matrix.

A downside of this method is that, in many practical situations, r might not be known *a priori*. Even worse, the presence of large corruptions in E can significantly compromise the estimation of L. It is possible to demonstrate that a single corrupted entry can induce PCA to estimate a solution that is arbitrarily far from the correct one [14]. Hence, in order to generate decompositions with a broader range of usability, more error-tolerant approaches must be considered.

2) Robust principal component analysis: In computer vision applications, it is desirable that the presence of some visual anomalies, like partially occluding objects, does not compromise the model accuracy. The robust PCA (RPCA) [14] presents an interesting solution to this issue by forcing a sparse error term E, that is, with most of its entries equal to zero.

To this end, Eq. (2) is modified so as to as generate the following minimization problem

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

where $||.||_0$ is the l_0-norm (number of non-zero entries), and the parameter λ balances the sparsity of E and the rank of L. Notice that, unlike the case of Eq. (2), the rank of L is not constrained, and should be considered an intrinsic property of the data.

Unfortunately, this problem is intractable due to its combinatorial nature, and its convex relaxation is considered instead [15]:

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

where $||.||_*$ is the *nuclear norm* of a matrix, defined by $||A||_* = \text{trace}(\sqrt{A^T A})$. The main advantage of this latter formulation is that it can be solved with high probability p under very weak conditions if λ is set to $1/\sqrt{\max(m,n)}$ [14], where m and n are the matrix dimensions.

The RPCA algorithm works very efficiently on scenarios with relatively static background, usually acquired by a static camera. According to Eq. (3), the method decomposes the background into the low-rank component L, while any other moving objects are isolated into E. This decomposition succeeds whenever the columns of L can be assumed to lie within a single subspace.

3) Principal subspace analysis: The pursuit for more general models has shown that the *union of subspaces* can be a more accurate representation of high-dimensional data when compared to the single subspace approach [16], [17]. This problem is clearly more complex than the single subspace modeling since it is difficult to determine, without any previous information, if a given sample is an outlier or it represents another subspace. The method known as *robust subspace recovery* (RoSuRe), that is described in the sequel, solves this by taking into account all samples simultaneously.

Consider $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k$ to be subspaces in $\mathbb{R}^m$ and let $L_1, L_2, \ldots, L_k$ be matrices where the columns of each L_j are vectors uniformly sampled from $\mathcal{S}_j$, assuming sufficient sampling density such that each column of L_j can be represented by the other columns with high probability. Since the above assumption is equivalent to saying that L_j is *self-representative*, there exists a square coefficient matrix W_j with null diagonal which allows one to write $L_j = L_j W_j$. Let $\bigcup \mathcal{S} = \bigcup_{j=1}^{k} \mathcal{S}_j$ be the union of the $\mathcal{S}_j$ subspaces. Therefore, it is fair to say that $L = [L_1|\dots|L_k]$ is a self-representative sample matrix of $\bigcup \mathcal{S}$ in the same way each L_j is for $\mathcal{S}_j$. Indeed, one can easily build a block-diagonal matrix $W = \text{diag}(W_1, \dots, W_k)$ which provides

$$L = LW, \tag{5}$$

where W has a null diagonal. By construction, W is expected to be sparse and it reveals the underlying subspace structure represented by L, which is said to be *blockwise low-rank* in a basis induced by L. So, apart from possible corruptions, recovering $\bigcup \mathcal{S}$ from the sampled data is equivalent to recovering L along with W. Thus, if X stacks n noisy observations of $\bigcup \mathcal{S}$, the subspace recovery can be obtained by performing the decomposition

$$X = LW + E, \tag{6}$$

where E is once again the matrix of perturbations.

The algorithm proposed in [18], [19] assumes the sparsity of both W and E matrices to obtain the representation in (6). In the same fashion as in RPCA, it does so by solving the relaxation of a harder combinatorial optimization problem. The resulting relaxed problem

$$\min_{W,E} ||W||_1 + \lambda ||E||_1 \quad \text{s.t.} \quad \begin{cases} X = L + E \\ LW = L \\ W_{ii} = 0 \end{cases}, \tag{7}$$

is not convex due to the bilinearity of W and E, but the global optimizer can be approximated by the *augmented Lagrangian multiplier* (ALM) method [19].

Apart from successful tests with synthetic data, the RoSuRe method also demonstrated strong potential in moving-camera surveillance problems [7], [20].

B. Domain transformations

The use of linear modeling techniques applied to images that are samples from a given process was an important breakthrough in image analysis. However, the success of these methods is strongly dependent on the pixelwise correlation among such images. It is known that even small misalignments among them can break the linear structure that is being modeled, compromising the low-rank assumption upon the data. However, unless sample images are previously registered or acquired under controlled conditions, geometric misalignments will occur, and are indeed very common in practice, specially when dealing with moving cameras. To work around this issue, images can be considered to lie in a different geometric domain, the misalignments being modeled as domain transformations. Many techniques try to model simultaneously the data while searching for the best domain transformation that optimizes its representation parsimony [13], [21], [22].

In this context, the robust alignment by sparse and low-rank (RASL) [13] decomposition tries to solve the RPCA problem with sample images that were not previously aligned. It works with batch alignments of linearly correlated images, instead of aligning single images.

Let $D = [I_1|\dots|I_n]$ be a matrix of observations, where each column is an image stacked into a flat vector. By general assumption, its samples are not aligned and are possibly corrupted. Let $\tau = [\tau_1|\dots|\tau_n]$ be a set of domain transformations that act on each sample of D, in such a way that

$$D \circ \tau = [I_1 \circ \tau_1|\dots|I_n \circ \tau_n], \tag{8}$$

where the resulting $I_j \circ \tau_j$ vectors may have a dimension that is different from the one of I_j. The entries of $D \circ \tau$ can be considered as selected regions of the entries of D that were geometrically transformed by each entry of τ. Suppose A is a matrix that, given a proper τ, contains the aligned entries of D. In this sense, A is approximately low-rank and thus one can write that $D \circ \tau = (A + E)$, where E encapsulates any possible data corruption. With this constraint, the RASL approach can be defined as the optimization problem

$$\min_{A,E,\tau} \text{rank}(A) + \gamma \|E\|_0 \quad \text{s.t.} \quad D \circ \tau = A + E. \tag{9}$$

In this framework, it is quite reasonable to assume that the best alignment of the samples in D minimizes the rank of A. Assuming the sparsity of the residue E, the solution of Eq. (9) simultaneously attempts to align and model the samples while compensating for the presence of any sparse corruptions. Eq. (9) can be solved by usual convex relaxation and the local linearization of τ in each iteration step [13].

Some recent approaches, namely [23], [24], [25], explore the use of domain transformations to cope with misalignment between temporally close frames in background subtraction applications. Although the results of these methods are encouraging, none of them explore the use of videos acquired by moving cameras. Due to the nature of the problem, change detection in moving-camera videos demands the ability of comparing frames acquired at different times and whose field-of-view (FoV) only partially overlap. Thus, unlike these previous publications, the use of domain transformations in this application requires the compensation of much more than slight frame misalignments due to a camera jitter.

III. Moving-Camera Surveillance with Sparse Representation

A. The moving-camera surveillance problem

Despite the success of several well established surveillance techniques using fixed cameras, the use of fixed-camera solutions can be expensive or unpractical in certain complex scenarios. Attaching a camera to a moving platform poses as an interesting work around to reach several viewpoints without increasing the number of cameras and, consequently, all the computational complexity related to them. This investigation addresses the problem of detecting changes in video sequences acquired by this kind of recording arrangements.

To describe the precise setup of the problem, some terminology is needed. Operator-validated sequences containing no anomalies are labeled as *reference videos*. These videos should be used to model the expected behaviour of the surveilled area. On the other hand, unsupervised video sequences that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2940686, IEEE Transactions on Image Processing

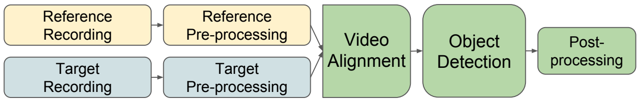


Fig. 1. The generic framework of anomaly detection with a moving camera.

should be inquired by the detection algorithm in order to locate any abnormal presence are called *target videos*. Since most moving-camera techniques perform a frame-by-frame comparison, temporal and geometric alignment between reference and target videos is often required. Image processing techniques may also be considered to reduce the effect of undesired artifacts (e.g. noise, sporadic bright spots, illumination normalization, etc.). Figure 1 summarizes the usual framework that is followed by most of the solutions.

A few recent works try to solve the moving-camera detection problem by a diverse range of techniques. Solutions like [9], [10] rely on feature-tracking to perform foreground/background separation. However, they are not suitable for reference/target videos comparison. The methods proposed in [6], [11] use image features to perform the geometric-registration step and employ the normalized cross-correlation (NCC) for actual anomaly detection. However, both of these works use external information to perform the temporal alignment, limiting their applicability range. Machine learning techniques like deep convolutional networks, dictionary learning of spatio-temporal features, and principal subspace analysis (PSA) are used respectively in [20], [26], and [27], with the latter showing the best results. However, due to camera vibration, such videos often contain significant geometric misalignments between corresponding target and reference frames, leading to a significant number false alarms in all these solutions.

In this paper we propose a new method to detect changes in moving-camera video sequences by performing a sparse representation of the target video in a transformed domain that copes with high levels of geometric misalignment with the reference video.

B. Adapting sparse representation to moving-camera surveillance

As mentioned before, linear approximation methods (Eqs. (1) to (7)) can be successfully employed on video surveillance problems to model video background using one or many low-rank subspaces and treating foreground anomalies as sparse outliers. However, dealing with backgrounds acquired with a moving camera can be significantly difficult as, due to the perspective effect, structures closer to the camera move faster along the video. This phenomenon can break the linearity assumption, “confusing” the modeling process and introducing undesired sparse non-linear artifacts in the corruption component.

Another important issue is the way modeling and detection are performed in such framework. For a static background, these two steps can easily be done simultaneously once a single sample contains information of the whole background and anomalies are not expected to be present in every sample, or at least not in the same spatial locations. In the case of a moving-camera background, several samples may be needed for representing each point-of-view. Additionally, the presence of outliers can induce the incorrect modeling of these objects as background if their relative frequency is high.

The moving-camera RoSuRe (mcRoSuRe) method [7] was proposed in order to work around these issues, and considers the modeling and detection stages into two different stages. First, a subspace learning stage, where the reference video is modeled, followed by a sparse representation stage, that isolates any anomaly present in the target video.

Let X_r be a data matrix which columns are composed by reference samples, generally frames from a reference video stacked as long column vectors. The modeling step uses the RoSuRe algorithm (Eq. (7)) to decompose X_r such that

$$X_r = L_r W_r + E_r, \tag{10}$$

where L_r is the blockwise low-rank part of the reference samples and is assumed to resemble the sampling of a union of linear subspaces, which structure is described in W_r. For this model, E_r is the corruption component, which is assumed to be sparse. Notice that E_r is not seen as a matrix of anomalies, but as residual information that could not fit the recovered model in L_r.

Now let X_t be a matrix of target samples, analogous to X_r. Assuming that X_t shares the same subspace structure with its reference counterpart, one can rewrite the blockwise low-rank part of X_t as a combination of the columns of L_r plus a sparse residual. In other words, one can find sparse matrices W_t and E_t, such that the target data matrix can be written as

$$X_t = L_r W_t + E_t. \tag{11}$$

Using this description, all anomalies present in X_t are encapsulated into E_t. To perform this alternate representation of X_t, taking advantage of L_r as determined in (10), a sparse representation algorithm inspired on RoSuRe is used, where the given low-rank term L_r is fixed. To do so, the new cost function is defined as

$$\min_{E,W} = ||W||_1 + \lambda||E||_1 \quad \text{s.t.} \quad X = LW + E, \tag{12}$$

where X and L are set up as input with X_t and L_r respectively.

The decomposition given in Eq. (11) tends to isolate in E_t all the target-sample information that is not present in L_r. However, besides the sparse corruptions generated by the anomalies, E_t will also have the residual sparse non-linear information that could not be captured by the blockwise low-rank $L_r W_t$ representation. The outlier information contained in E_t can be separated from its inherent non-linear residual by noting that, as X_t and X_r are similar by assumption, E_t will look in general quite similar to E_r, except around these anomalies. Therefore, in this method a third and last step is also performed, decomposing E_t using E_r as the input parameter L of the optimization described in Eq. (12), yielding

$$E_t = E_r W + E_e. \tag{13}$$

In the above equation, the remaining sparse component E tends to contain, as desired, just the outliers in X_t not present in X_r.

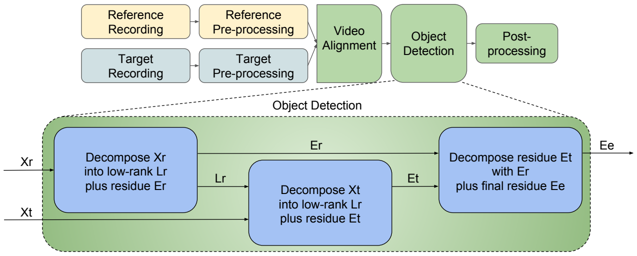


Fig. 2. mcRoSuRe optimization workflow inserted in the traditional moving camera object detection framework.

Such a method has the great advantage of obviating the necessity of geometric registration of each frame. This is extremely useful when the perturbations of the camera movement are not so large, or even in synthetic moving video like “virtual pannings” generated from pan-tilt-zoom (PTZ) cameras or alike, saving a considerable computational time.

Figure 2 summarizes the mcRoSuRe method optimization workflow.

IV. Domain-Transformable Sparse Representations

Recent state-of-art techniques, such as the mcRoSuRe algorithm briefly described in Section III, make use of subspace learning to find a low-rank representation model of the moving-camera reference sequence. For small perturbations in the camera path, this technique can successfully extract the target video anomalies, exploiting the strong correlation between consecutive frames in both the reference and target videos, and also between the corresponding frames in these two sequences.

However, in many moving-camera practical scenarios, it is quite difficult to avoid eventual geometric misalignments between corresponding frames. Cameras attached to moving platforms may suffer from this problem since these devices may have to deal, besides normal camera vibration, with path irregularities or unpredicted weather conditions, for example.

In this section we discuss the use of geometric domain-transformations to model the effect over the samples caused by the camera movement and rotation referred above. In short, the idea is to include a transformation term in the sparse representation procedure presented in Eq. (12), updating the optimization algorithm to reflect these changes.

Let $X_r \in \mathbb{R}^{m\times n_r}$ and $X_t \in \mathbb{R}^{m\times n_t}$ be the reference and target matrices, respectively, containing information about their corresponding video sequences. Let m be the number of pixels in the video frames, and n_r and n_t the number of frames in the reference and target videos, respectively. In a first moment, let us assume a scenario where both reference and target sequences were acquired with similar camera path and pose, as considered in [6], [11], [26]. Under these conditions, it is fair to assume that every column in X_t has at least one corresponding column in X_r, leading to the problem given by

$$\min_{W,E} \|W\|_1 + \lambda\|E\|_1, \quad \text{s.t.} \quad X_t = X_r W + E, \tag{14}$$

where $W \in \mathbb{R}^{n_r\times n_t}$ is a coefficient matrix, which describes the relations of the columns of X_t and X_r, and $E \in \mathbb{R}^{m\times n_t}$ is the error term (that arises from the mismatches between X_r and X_t), which has the same dimensions of X_t. In this problem, the factor λ is used to balance the importance of the two minimized terms and may be adjusted considering the expected amount of sparsity in W and E. Notice that Eq. (14) is basically Eq. (12), but under a different context where the representation is direct, and not made on top of a subspace model.

Now, let us relax the previous assumptions by breaking the requirement that X_t and X_r are perfectly aligned, since there are uncontrolled camera shaking and rotation. Since images acquired at the same center of projection (which is now our only assumption) can be related by a homography transformation [28], it is possible to consider that the target observations in X_t are in a different geometric domain with respect to X_r. This assumption allows one to model the camera shaking as a geometric transformation applied to a domain where corresponding target and reference samples are aligned. In this sense, Eq. (14) becomes

$$\min_{W,E,\tau} \|W\|_1 + \lambda\|E\|_1, \quad \text{s.t.} \quad X_t \circ \tau = X_r W + E, \tag{15}$$

where $\tau = [\tau_1 \ldots \tau_{n_t}]$ is an vector of domain-transformations, in the same fashion as the discussed in Section II-B. Eq. (15) generalizes the model considered in Eq. (14) by incorporating the transformation τ. The additional degrees of freedom inherent to τ allow a better match between X_r and X_t, minimizing the content of E as desired. Each entry of τ acts on its corresponding column of X_t, that represents the observed target samples. For implementation purposes, τ is represented by a $p \times n_t$ matrix where each column is a vector with the p parameters necessary to describe each transformation acting on X_t.

Although at a first glance one might think this development is just a new implementation of [13], since it appears that the domain transformations are used in the same way, a thorough inspection of the proposed formulation shows that both X_r and X_t are fixed in the present formulation. Also, it would come naturally from [13] that the X_r matrix would be modified by the transforms, since it plays a similar logical role in the present work and in [20] as the X matrix in [13]. We have chosen, however, to apply the transformations over X_t, since we assume here that X_r is well known by the system and is considered to be the best representation available to the background model. As for the X_t matrix we assume it might suffer from misalignments that should be corrected before the decomposition is performed.

In this problem, the composition of a geometric domain-transformation with X_t breaks the linearity of the optimization constraint that appears on the right side of Eq. (15). However, in order to iterate on τ along the optimization procedure, one can perform a first order approximation. Hence, we consider that, for a small variation $\Delta\tau = [\Delta\tau_1|\ldots|\Delta\tau_{n_t}] \in \mathbb{R}^{p\times n_t}$ of τ, it is possible to approximate this constraint by linearizing $X_t \circ \tau$ with the current estimate of τ, in a similar way it is done in [21] and [22], such that

$$X_t \circ (\tau + \Delta\tau) \approx X_t \circ \tau + \sum_{i=1}^{n_t} J_i \Delta\tau \epsilon_i \epsilon_i^\top, \tag{16}$$

where the ϵ_i represent the canonical basis for $\mathbb{R}^{n_t}$ and

$$J_i = \frac{\partial}{\partial \zeta} \left(\frac{(X_t)_i \circ \zeta}{\|(X_t)_i \circ \zeta\|_2} \right) \bigg|_{\zeta = \tau_i} \in \mathbb{R}^{m \times p} \tag{17}$$

is the Jacobian of the i-th column of the target matrix X_t with respect to τ_i.

This leads to the modified optimization problem

$$\begin{aligned} &\min_{W,E,\Delta\tau} \|W\|_1 + \lambda \|E\|_1, \\ &\text{s.t.} \quad X_t \circ \tau + \sum_{i=1}^{n_t} J_i \Delta\tau \epsilon_i \epsilon_i^\top = X_r W + E. \end{aligned} \tag{18}$$

Hence, the solution of Eq. (15) can be achieved by repeatedly solving (18) and then updating τ at each iteration step. In practice, the whole optimization of the so-called moving-camera domain-transformation sparse representation (mcDTSR) algorithm can be computed within two loops.

In the outer loop, the Jacobian matrices J_i are computed based on the current estimate of τ. Then, the columns of $X_t \circ \tau$ are normalized to avoid undesired trivial solutions, like, for example, zooming to a black pixel of a given frame of X_t, that will end up with a null column of W. This explains why the derivative in Eq. (17) is also normalized. Only after that the inner loop is performed by solving (18). At last, τ is updated by the $\Delta\tau$ increment that is recurrently computed in the inner loop, which also estimates the sparse coefficient W and the error E. We have empirically observed that the relative change of the objective function is a good stopping criterion for the outer loop, meaning that given a positive value ε_r, the outer loop is exited when

$$\frac{|\text{obj}_k - \text{obj}_{k-1}|}{|\text{obj}_k|} < \varepsilon_r, \quad \text{obj}_k = \|W_k\|_1 + \lambda \|E_k\|_1, \tag{19}$$

where k is the iteration index of the outer loop. In this procedure, an initial set of transformations τ_0 must be provided for the outer loop, along with the X_r and X_t matrices. If one assumes that both of these matrices were acquired under similar conditions, τ_0 can be initially chosen as a set of identity transforms. The proposed method for the mcDTSR outer loop is summarized in Algorithm 1. To efficiently solve Eq. (18) inside the inner loop, we shall make use of the augmented Lagrangian method [15]. By defining

$$h(W, E, \Delta\tau) = X_t \circ \tau + \sum_{i=1}^{n} J_i \Delta\tau \epsilon_i \epsilon_i^\top - X_r W - E, \tag{20}$$

one can write the augmented Lagrangian function as

$$\begin{aligned} \mathcal{L}_\mu(W, E, \Delta\tau, Y) = {} & \|W\|_1 + \lambda \|E\|_1 + \langle Y, h(W, E, \Delta\tau) \rangle \\ & + \frac{\mu}{2} \|h(W, E, \Delta\tau)\|_F^2, \end{aligned} \tag{21}$$

where Y is Lagrange multiplier matrix and μ is a positive scalar. This can be solved by estimating both Y and the optimal solution iteratively [29] as follows

$$\begin{aligned} (W_{k+1}, E_{k+1}, \Delta\tau_{k+1}) &= \arg \min_{W,E,\Delta\tau} \mathcal{L}_{\mu_k}(W, E, \Delta\tau, Y_k), \\ Y_{k+1} &= Y_k + \mu_k h(W_{k+1}, E_{k+1}, \Delta\tau_{k+1}), \\ \mu_{k+1} &= \rho \mu_k, \end{aligned} \tag{22}$$

where μ_0 and ρ are tunable parameters that will be discussed later. To facilitate the solution of Eq. (22), we can break it into three and approximate the result by minimizing one unknown at a time, such that

$$\begin{aligned} W_{k+1} &= \arg \min_{W} \ \mathcal{L}_{\mu_k}(W, E_k, \Delta\tau_k, Y_k), \\ E_{k+1} &= \arg \min_{E} \ \mathcal{L}_{\mu_k}(W_{k+1}, E, \Delta\tau_k, Y_k), \\ \Delta\tau_{k+1} &= \arg \min_{\Delta\tau} \ \mathcal{L}_{\mu_k}(W_{k+1}, E_{k+1}, \Delta\tau, Y_k). \end{aligned} \tag{23}$$

The great advantage of alternating the unknowns in (23) is that each one has a direct form of computation.

As in [18], W and E can be estimated by the soft-thresholding operator, defined as

$$\mathcal{S}_\gamma[A] = \operatorname{sign}(A) \cdot \max\{|A| - \gamma, 0\}, \tag{24}$$

where the sign and max operations are applied entrywise on the matrix A.

Expanding the expressions in Eq. (23) using the same rationale underlying the development in [19], we would have:

$$\begin{aligned} W_{k+1} = \mathcal{S}_{\frac{\lambda}{\mu_k}} \Bigg[& W_k - X_r^\top \Big(X_r W_k + E_k - \\ & -X_t \circ \tau - \textstyle\sum_{i=1}^n J_i \Delta\tau_k \epsilon_i \epsilon_i^\top + \frac{1}{\mu_k} Y_k \Big) \Bigg], \\ E_{k+1} = \mathcal{S}_{\frac{\lambda}{\mu_k}} \Bigg[& E_k - \Big(X_r W_{k+1} + E_k - \\ & -X_t \circ \tau - \textstyle\sum_{i=1}^n J_i \Delta\tau_k \epsilon_i \epsilon_i^\top + \frac{1}{\mu_k} Y_k \Big) \Bigg], \\ \Delta\tau_{k+1} = \mathcal{S}_{\frac{\lambda}{\mu_k}} \Bigg[& \Delta\tau_k - \mathcal{C}^* \Big(X_r W_{k+1} + E_{k+1} - \\ & -X_t \circ \tau - \textstyle\sum_{i=1}^n J_i \Delta\tau_k \epsilon_i \epsilon_i^\top + \frac{1}{\mu_k} Y_k \Big) \Bigg], \end{aligned} \tag{25}$$

where $\mathcal{C}^*(\theta)$ is the adjoint of the functional $\mathcal{C}(\theta) = \sum_{i=1}^n J_i \theta \epsilon_i \epsilon_i^\top$ which is applied over $\Delta\tau$ in $h(W, E, \Delta\tau)$.

However, in our application the $\Delta\tau$ is not assumed to be sparse, therefore we chose not to apply the soft threshold operator $\mathcal{S}_\gamma[\cdot]$ in its update equation, replacing the last line of Eq. (25) by:

$$\begin{aligned} \Delta\tau_{k+1} = {} & \Delta\tau_k - \mathcal{C}^* \Big(X_r W_{k+1} + E_{k+1} - \\ & -X_t \circ \tau - \textstyle\sum_{i=1}^n J_i \Delta\tau_k \epsilon_i \epsilon_i^\top + \frac{1}{\mu_k} Y_k \Big). \end{aligned} \tag{26}$$

In our application, the functional $\mathcal{C}^*(\theta)$ is defined by $\sum_{i=1}^n J_i^{-1} \theta \epsilon_i \epsilon_i^\top$.

Since the space-size parameter p is relatively small when compared to the frame resolution dimension m, the Jacobian matrices J_i are likely to be ill-conditioned, which may lead to numerical instability in the inner loop. To work this around, one may perform a QR factorization of the Jacobians, that is, $J_i = Q_i R_i$, and use orthogonal factors Q_i inside the inner loop in place of the Jacobians J_i. In this manner, the inner loop

Algorithm 1 - Domain-transformable sparse representation for moving camera videos (mcDTSR): outer loop

```
input: Reference matrix $X_r \in \mathbb{R}^{m \times n_r}$, target matrix $X_t \in \mathbb{R}^{m \times n_t}$, inital transformation vector $\tau = [\tau_1 \dots \tau_{n_t}] \in \mathbb{R}^{p \times n_t}$, and weight $\lambda > 0$

while not converged (Eq. (19) is not satisfied) do
    (step 1) compute Jacobian matrices for each $\tau_i$
        $J_i \leftarrow \frac{\partial}{\partial \zeta} \left( \frac{(X_t)_i \circ \zeta}{\|(X_t)_i \circ \zeta\|_2} \right) \Bigg|_{\zeta = \tau_i}, \quad \text{for } i = 1, \dots, n_t;$
    (step 2) warp and normalize the images in $X_t$ matrix
        $X_t \circ \tau \leftarrow \left[ \frac{(X_t)_1 \circ \tau_1}{\|(X_t)_1 \circ \tau_1\|_2} \ \dots \ \frac{(X_t)_{n_t} \circ \tau_{n_t}}{\|(X_t)_{n_t} \circ \tau_{n_t}\|_2} \right];$
    (step 3) solve linearized convex optimization (inner loop):
        $(W^*, E^*, \Delta\tau^*) \leftarrow \arg \min_{W, E, \Delta\tau} \|W\|_1 + \lambda \|E\|_1 \quad \text{s.t.}$
        $X_t \circ \tau + \sum_{i=1}^{n_t} J_i \Delta\tau \epsilon_i \epsilon_i^\top = X_r W + E.$
    (step 4) update the transformation vector:
        $\tau \leftarrow \tau + \Delta\tau$
    .
end

ouput: solution $W^*$, $E^*$, and $\tau^*$ to problem (15).
```

will output $\overline{\Delta\tau_i} = R_i \Delta\tau_i$ instead of $\Delta\tau_i$ for each component of $\Delta\tau$, also the inner loop will only see the Q_i components of each J_i. Since the R_i are invertible, $\Delta\tau$ can be easily computed [13].

The mcDTSR inner loop described in Algorithm 2 solves separately for both W and E, using the linearized alternating direction method with adaptive penalty (LADMAP) approach [29], differently from the approach in [13], where the ALM is applied. By expanding the Lagrangian using LADMAP one is able to reach a faster convergence [29]. The use of LADMAP is the reason μ_k is updated by a positive ρ (Eq. (22)). The value of ρ has influence on the compromise between approximation accuracy and the algorithm's running time. For the stopping criterion of the inner loop, one may consider the ratio between the Frobenius norm of h (which can be thought of as the residual of the cost function in Eq. (18)), and the norm of $X_t \circ \tau$ itself. More precisely, the inner loop will stop when

$$\frac{\|h(W_k, E_k, \Delta\tau_k)\|_F}{\|X_t \circ \tau\|_F} < \varepsilon_t, \tag{27}$$

where k is the current inner-loop iteration index.

Assuming that reference and target videos may have significant misalignments between its correspondent frames, working with the full video frames can make the warped frames of X_t present invalid pixels at the borders, that are the result of the mapping of pixels beyond the borders of X_t. As these invalid pixels can affect the algorithm convergence, a common practice is to work with a region-of-interest (ROI) window that is smaller than the full video frame, so that it can have some freedom to warp and avoid the mapping of pixels outside the frame's borders. Thus, X_t and X_r are in general ROI windows inside the full frames.

Figure 3 shows the block diagram for the proposed algorithm.

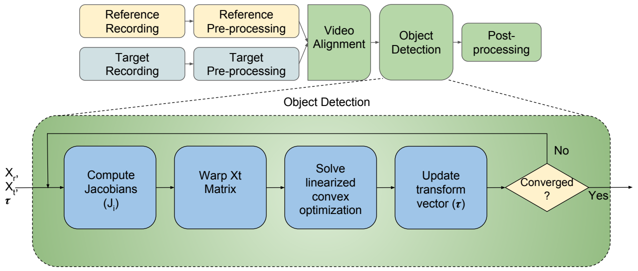


Fig. 3. mcDTSR block diagram.

V. Experimental Results

A. Testing dataset

In order to assess the detection quality of our proposed technique we considered the VDAO [30] database, which contains several recordings on a complex industrial-like environment. The dataset sequences were acquired with a rigid camera mounted on a robotic iRobot$^{\text{TM}}$ *Roomba* platform with a back-and-forth linear movement along a fixed 6m-long hanging rail. Two different IP cameras were employed, having the same 1280×720 pixel resolution and frame rate of 24 fps. An industrial environment was considered, comprised

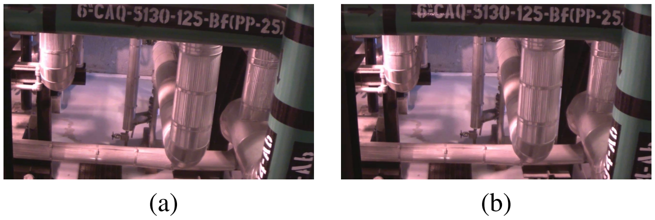
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2940686, IEEE Transactions on Image Processing

Algorithm 2 - Domain-transformed sparse representation for moving camera videos (mcDTSR): inner loop

```
input: $W_0 \in \mathbb{R}^{n_r \times n_t}$, $E_0 \in \mathbb{R}^{m \times n_t}$, $Q$, $\overline{\Delta\tau}_0 = 0 \in \mathbb{R}^{p \times n_t}$, $\mu_0 > 0$, $\rho > 0$, $\lambda > 0$

while not converged (Eq. (27) is not satisfied) do
    $W_{k+1} = \mathcal{S}_{\frac{\lambda}{\mu_k}} \left[ W_k - X_r^\top \left( X_r W_k + E_k - X_t \circ \tau - \sum_{i=1}^{n} Q_i \overline{\Delta\tau}_k \epsilon_i \epsilon_i^\top + \frac{1}{\mu_k} Y_k \right) \right]$;
    $E_{k+1} = \mathcal{S}_{\frac{\lambda}{\mu_k}} \left[ E_k - \left( X_r W_{k+1} + E_k - X_t \circ \tau - \sum_{i=1}^{n} Q_i \overline{\Delta\tau}_k \epsilon_i \epsilon_i^\top + \frac{1}{\mu_k} Y_k \right) \right]$;
    $\overline{\Delta\tau}_{k+1} = \overline{\Delta\tau}_k + \sum_{i=1}^{n} Q_i^\mathrm{T} \left( X_r W_{k+1} + E_{k+1} - X_t \circ \tau - \sum_{j=1}^{n} Q_j \overline{\Delta\tau}_k \epsilon_j \epsilon_j^\top + \frac{1}{\mu_k} Y_k \right) \epsilon_i \epsilon_i^\top$;
    $Y_{k+1} = Y_k + \mu_k h(W_{k+1}, E_{k+1}, \Delta\tau_{k+1})$;
    $\mu_{k+1} = \rho \mu_k$;
end while

$\forall_i : \Delta\tau_i = R_i^{-1} \overline{\Delta\tau}_i$

ouput: solution $W^*$, $E^*$, and $\Delta\tau^*$ to problem (18).
```

of several pipes and valves, and 24 distinct abandoned objects were employed in the recordings, which total approximately 8.2 hours of annotated video. To the best of our knowledge, at the time of writing, the VDAO was the only publicly available dataset designed for object-detection in moving-camera video sequences, as indicated by [31]. Even other recent object-detection surveys such as [32] do not feature similar moving-camera datasets comprising reference (without objects) and target (potentially with objects) videos. In fact, some of the more broadly used anomaly detection datasets from recent years such as [33], [34] feature only a limited amount of moving-camera surveillance videos without, however, reference and target corresponding pairs. Other object-detection works such as [35], [36], although featuring over 6 different datasets in their experiments, do not include anyone designed in a similar way as that of the VDAO database [30].

In the VDAO database, the recordings were divided into two groups: reference and target sequences. The reference sequences have no abandoned objects, as validated by human supervision, while the target sequences contain one or more objects to be detected automatically by the proposed mcDTSR algorithm. Due to track imperfections and mechanical friction with the robot wheels, the captured sequences present considerable camera trepidation. These camera trepidations and jitter cause the images from reference and target videos not to match perfectly. Even when comparing frames acquired by the camera while the robot is at the same position at two different instants, the fields-of-view of the camera at different posisions do not overlap completely, making the database a very challenging one. Among the challenges presented by the VDAO database are the temporal and geometric misalignments that make this database even more complex. As stated before, there are different camera poses between corresponding reference and target frames acquired in the same rail position, as illustrated in Figure 4. This effect hinders the ability of traditional algorithms to find the correct reference-target frame match. Other VDAO challenges include object occlusions and the fact that objects in different depths may appear differently due to some parallax effect caused by camera rotations. By considering all these issues, the VDAO dataset allows one to test his/her algorithm in a quite challenging anomaly-detection scenario. This database, along with the ground truth annotations of the abandoned objects, can be downloaded from [37].



(a) (b)

Fig. 4. Example of geometrical mismatch between reference and target frames of the VDAO dataset: (a) reference frame; (b) target frame. One can notice the camera rotation between the two frames, that was caused by different camera poses during the video acquisition.

A special selection of the database called VDAO-200 [38] is used to perform the qualitative and quantitative experiments in upcoming Sections V-C and V-D, respectively. This auxiliary database is composed of 59 excerpts with 200 frames taken from VDAO single-object target videos. The selection contains a total of 9 different objects in different positions and 2 types of illumination. On almost half of the videos, the objects are partially or completely occluded. There are also several situations of environment shadow casting, different object shapes and camera shaking, which make the selection very challenging and also representative of the full database.

B. Domain-transformation compensation

The VDAO database comes with ground truth annotations of the abandoned objects for every target-video frame, where the object positions are marked with rectangular bounding boxes. Since the abandoned objects have arbitrary shapes, working with bounding boxes can lead to results which are not very precise, that may mask the actual amounts of true and false positives.

This said, another relevant concern is that τ is computed with respect to the target video, so any evaluation metric should take into account the domain-transformation performed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2940686, IEEE Transactions on Image Processing

on each frame of X_t. Since a general transformation should change the annotated bounding boxes into quadrilateral polygons, we chose to maintain the target domain fixed and apply the inverse transformation τ^{-1} to the reference domain when carrying out the performance assessment. More precisely, applying τ^{-1} to both sides of the constraint in Eq. (15) leads to

$$X_t = X_r W \circ \tau^{-1} + E', \tag{28}$$

where $E' = E \circ \tau^{-1}$. To compute E' we consider the transformation applied to the whole image, and not only the region-of-interest. So, a general transformation on $X_r W$ may yield frames that contain zero values near their boundaries, since the image border may be overlapped by the resulting quadrilateral. To avoid dummy false positives, E' is set to zero in these problematic regions. The metrics described in the sequel are applied to this resulting error image, after all post-processing steps. These choices are motivated by simplicity and the possibility to compare our results with the ones other methods with no need for any adaptation of these other results.

In Section IV we discussed that a possible initialization for the τ_0 parameter could be the identity matrix. If in a given video the geometric misalignment is large enough, however, this could result in unnecessary costly iterations due to a bad transformation initialization. To manage this issue, one can run the outer loop optimization for a single (possibly central) frame and obtain an “initial guess” for the geometrical transformation $\tau = [\tau_1 \dots \tau_{n_t}] \in \mathbb{R}^{p \times n_t}$ for all other video frames, as given in Algorithm 1.

To assess how the use of such initialization method would impact the performance of the algorithm we ran two versions of the code: first with the τ_0 initialized as an identity matrix and later with the proposed “initial guess”. We compared both initializations in terms of number of outer loop iterations (Algorithm 1) and processing time (already considering the time used to compute the initial guess). For this test we employed a computer with an Intel i7-4712HQ processor at 2.4Hz and 16 GB of RAM. Table I shows the results for both methods using all the 59 videos from VDAO-200 dataset.

TABLE I
PERFORMANCE TEST OF THE TWO PROPOSED τ_0 INITIALIZATION SCHEMES.

Initialization	Total Iter.	Avg. Iter.	Total Time (s)	Avg. Time (s)
Identity	1404	23.80	1506879	25540.32
Initial Guess	966	16.37	1067791	18098.15

By inspecting Table I one can readily see that the proposed “initial guess” transformation accelerates the algorithm reducing the number of the outer loop iterations by over 31% and the total time by more than 27%. This comes with virtually no difference in the algorithm detection performance. Therefore, for the remainder of the experiments, the proposed “initial guess” initialization is employed.

C. Qualitative evaluation

In this section, we illustrate the advantage of including domain transformations into the optimization process. When the corresponding target and reference sequences have considerable levels of misalignment, the sparse representation of the target frames performs poorly, generally introducing several artifacts into the residual component E. If some algorithm that uses low-rank or sparse representation is used for detection purposes, this misalignment can yield a large number of false positive regions, possibly masking the actual presence of strange objects on the scene, compromising the practical applicability of such a method. In this sense, a simple experiment was designed to illustrate and qualitatively evaluate the gain in detection performance provided by the proposed algorithm. To this end, the main components of Eq. (18) will be inspected along the iterations of the mcDTSR outer loop described in Algorithm 1, providing some insights about what is happening “under the hood”.

For this task, we have selected an excerpt of the target video from the VDAO database entitled “Object 3 (shoe, position 3)”. This sequence presents a case of significant misalignment with respect to its corresponding reference video, making any conventional method that is not tolerant to camera shaking not to perform well. A 50-frame snippet of this target video was selected together with a 100-frame snippet of the corresponding reference video, manually chosen such that the entire target excerpt can be represented by the reference one. It is important to point out that, although in this case the target-reference match is guaranteed, the algorithm has no information about which reference frames shall be used to represent an arbitrary target frame, nor about the parameters of camera tilt between these corresponding frames. To reduce the processing time, these video snippets where downsampled to a 320×180-pixel resolution and converted to grayscale, and the chosen regions-of-interest (ROI) were the 280×150-pixel central windows from each frame in both videos.

In theory, any parametrizable geometrical transform could be used as τ, with an unlimited number of degrees of freedom for the deformations applied to the target frames. However, considering our target application, we chose to consider that any two corresponding frames could be matched through planar homographies. By considering only these transforms to represent the domain transformation, one gets $p = 8$ by using a 4-point parametrization to describe the columns of τ. An example of the frame matching achieved by the use of such planar homographies can be seen in Figure 5, where one can notice that the applied transform corrected the position of the target frame that after the transformation approximates the reference frame.

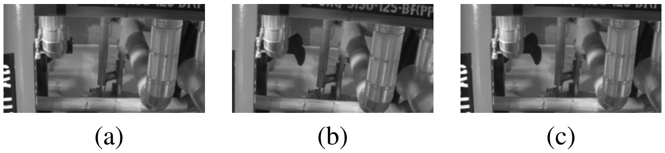
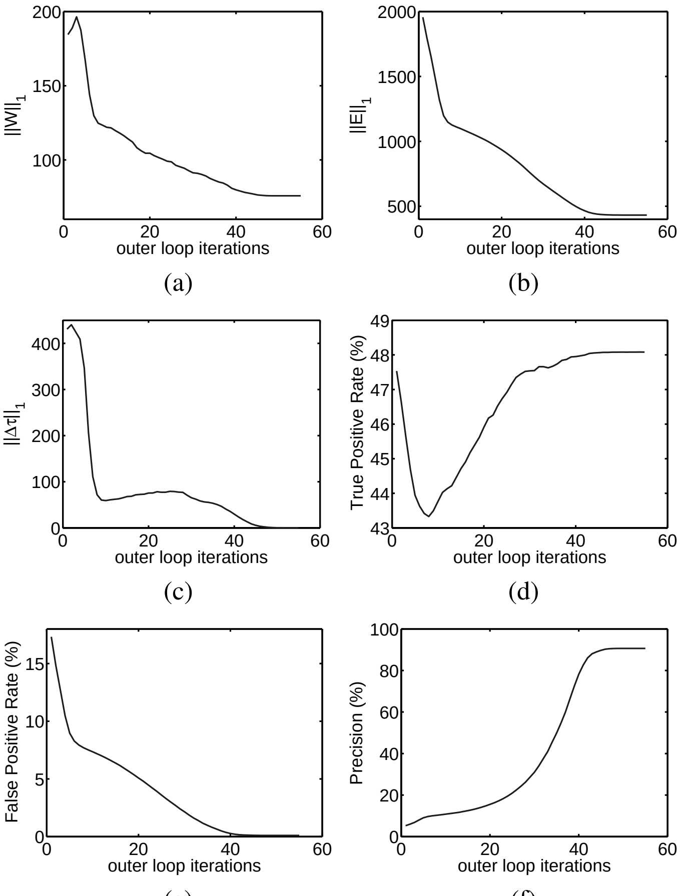

(a) (b) (c)

Fig. 5. Example the homography transformation applied to a target frame: (a) reference frame; (b) target frame; (c) transformed target frame. One can notice that the original target frame presented a geometrical mismatch when compared with the reference frame. This mismatch is corrected when a homography transformation is applied to the target frame, resulting in a registered image.

The parameter setup used in mcDTSR was $\lambda = 10^2$, $\rho = 1.01$, and $\mu_0 = 1.25/\|X_t \circ \tau\|$, following the values used in [7]. The inner loop tolerance for the stopping criterion was set to $\varepsilon_t = 10^{-4}$ and the outer loop tolerance ε_r was left loose. The idea was to observe how the magnitudes of the algorithm unknowns and metrics behave along a total of 55 outer-loop iterations.

At the post-processing detection stage, a simple thresholding procedure was performed by marking as foreground every entry of $|E|$ with intensity greater than $\beta = 0.125$, otherwise turning it as background.



(a) (b)

(c) (d)

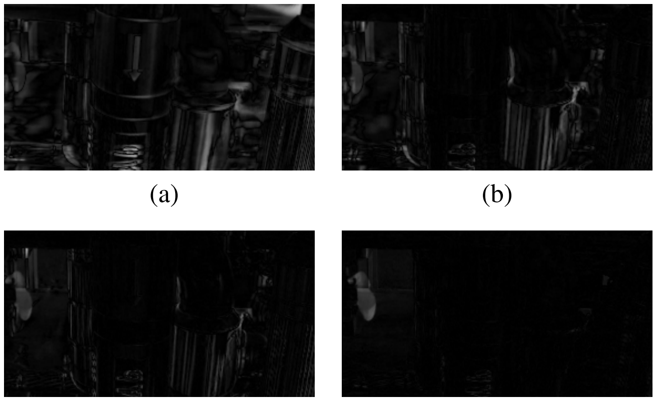
(e) (f)

Fig. 6. Evolution of several mcDTSR parameters and performance metrics along outer-loop iterations, illustrating improvement over time of proposed algorithm: (a) $\|W\|_1$; (b) $\|E\|_1$; (c) $\|\Delta\tau\|_1$; (d) True-positive detection rate; (e) False-positive detection rate; (f) Precision rate.

In Figs. 6(a), 6(b), and 6(c), it is possible to see the evolution of the l_1-norm of $\|W\|$, $\|E\|$, and $\|\Delta\tau\|$, respectively, across the outer-loop iterations. Independently of their magnitude ranges, one can clearly notice how the three norms evolve in time, converging to their final values after approximately 45 iterations.

However, the great strength of the proposed method can be noticed in Figs. 6(d), 6(e), and 6(f), where some detection metrics for the mcDTSR algorithm are displayed. All these three metrics were computed pixelwise, by comparing the binary mask video E', as given in Section V-B, to the provided bounding-box ground truth from the VDAO database.

The behaviour shown by the true-positive rate (TPR) plot in Fig. 6(d) is explained by the fact that the ground truth bounding boxes are larger than the actual object. Thus, this plot represents the superposition of the actual false positives that lie inside the bounding box being eliminated, promoting a decrease in the TPR, plus the actual object being increasedly detected. This can be appreciated by looking also at evolution of E over the outer-loop iterations in Fig. 7. This figure also explains the impressive false-positive rate (FPR) and precision plots depicted in Fig. 6(e) and Fig. 6(f), respectively, as a result from the improved geometric alignment between the target and reference frames. In fact, from the first outer-loop iteration ($i = 1$) to the last one ($i = 55$), more than 99% of the false positives were eliminated.



(a) (b)

(c) (d)

Fig. 7. Evolution of residual matrix $|E|$ through selected mcDTSR outer-loop iterations i, for a fixed video frame: (a) $i = 1$; (b) $i = 19$; (c) $i = 37$; (d) $i = 55$. Notice that the gradual alignment between target and reference correspondences contributes for an impressive reduction of potential false-positives regions, and also for a more precise detection of the abandoned object.

The geometric alignment can be crucial to the convergence of sparse representation methods. This is well illustrated by Figs. 8 and 9, which show the evolutions of the target ROIs and the estimated W, respectively. In fact, the improved geometric alignment provided by the transformation τ, as given in Fig. 8, enables a more robust and consequently more precise matrix factorization for the target video, as seen in Fig. 9.

D. Quantitative evaluation

For this experiment, we consider all 59 200-frame videos excerpts from the VDAO-200 subset, as given in Section V-A. The parameter setup for the proposed mcDTSR algorithm are the same as in Section V-C, with addition of the stopping criterion set to $\varepsilon_r = 10^{-5}$. The post-processing detection stage is composed by a thresholding step on $|E'|$ with $\beta = 0.2$, followed by morphological open and then close operations with 2 and 4 pixel-wide, disk-shaped structuring elements, respectively. This is followed by a simple temporal voting using a 5 pixel-wide window, that turns the pixel on if more than half of the window is also on.

To assess the performance of the proposed mcDTSR method, the following metrics are employed: true positive rate (TP) and false positive rate (FP). A true positive happens

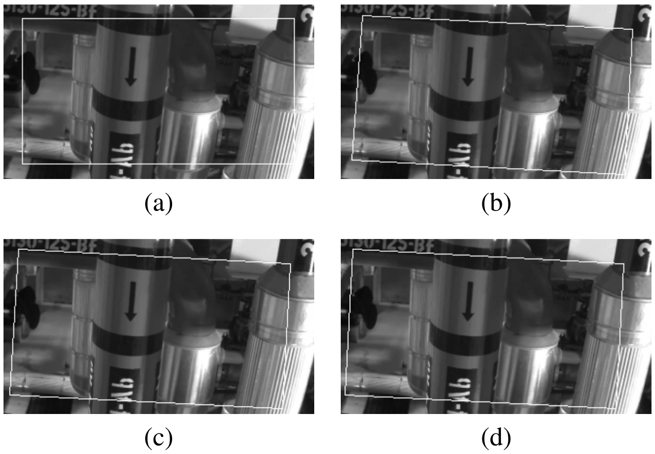


Fig. 8. ROI evolution through selected mcDTSR outer-loop iterations i, for a fixed video frame: (a) $i = 0$ (initial ROI); (b) $i = 19$; (c) $i = 37$; (d) $i = 55$.

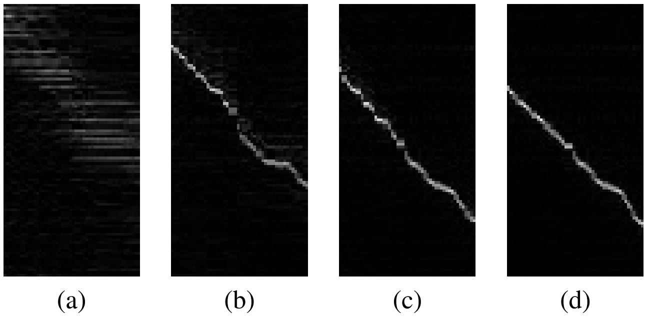


Fig. 9. Evolution of weight matrix $|W|$ through selected mcDTSR outer-loop iterations i, for a fixed video frame: (a) $i = 1$; (b) $i = 19$; (c) $i = 37$; (d) $i = 55$. Notice how the target frames are incorrectly temporally correlated with the reference frames at the first iterations. The optimization gradually shifts the correlation to the correct frames, thanks to the transformations applied to the target video.

when the detection blob has a non-empty intersection with the abandoned-object ground-truth bounding box, and a false positive happens when the detection blob and the ground-truth bounding boxes are disjoint. Another metric used for performance assessment is the DIS metric, that integrates both TP and FP, defined as

$$\mathrm{DIS} = \sqrt{(1 - \mathrm{TP})^2 + \mathrm{FP}^2}. \tag{29}$$

The DIS metric represents the minimum distance of the (TP,FP) point to the point of ideal behaviour (TP = 1 and FP = 0) on the TP×FP plane. The use of this metric allows direct comparison with the results in [6] and [20] for several state-of-the-art methods found in the literature, namely: the spatio-temporal composition for moving-camera detection (STC-mc) [27]; the detection of abandoned objects with a moving camera (DAOMC) [11]; the moving-camera background subtraction (MCBS) [26] the anomaly detection with a moving camera using multiscale video analysis (ADMULT) [6]; and the anomaly detection in moving-camera video sequences using principal subspace analysis (mcRoSuRe-A) [20]. The overall results for all these methods, including the proposed mcDTSR algorithm, are given in Table II.

The analysis of the results presented in Table II shows that the proposed mcDTSR method outperforms the state-of-the-art algorithms in 43 of the 59 videos, while also pairing up in 4 other videos. This shows that mcDTSR has superior individual performance over the other algorithms, but also that using domain transformations to deal reference/target misalignments is an improvement over mcRoSuRe-A, which is also a sparse representation technique. The average results of Table I can be seen in Table III, which shows that mcDTSR significantly reduces the average DIS score. This confirms the effectiveness of the introduction of the domain transformations in the detection pipeline.

It can be argued that object-level detection is a very harsh metric in applicability terms, so one may consider a less strict metric for detection performance based on a frame-level analysis, which does not specifically consider the anomaly position in the given frame. In this frame-level context the TP$_{fl}$ metric is affected by the presence of any blob detected in an anomalous frame. Conversely, the FP$_{fl}$ is determined by the presence of any blob detected in a non-anomalous frame. The average results for the frame-level comparison over the same 59 VDAO-200 videos is summarized in Table IV. These results confirm that mcDTSR has a clear performance advantage when compared to the STC-mc, DAOMC, MCBS, and ADMULT methods, and, more importantly, it represents a significant detection improvement when compared directly to the mcRoSuRe-A, even in a frame-level analysis, as can be seen in Tabel IV.

The post-processing setups for the mcDTSR and mcRoSuRe-A algorithms, used to obtain the results shown in Tables II,III,IV, were adjusted by a simple methodology. A parameter grid search is performed on 28 of the 59 videos, in order to minimize the object-level DIS score. The best parameter setup is then used to compute the results for all 59 videos in all three tables. In that search, diameters of the structuring elements of the morphological open and close range, each one, from 1 to 5; the threshold ranges from 0.2 to 0.3 in 0.01-steps, and the temporal voting windows are tested for the 3, 5, and 7 sizes. Although the same tuning methodology was used for both algorithms, the best setup of each one is chosen independently, yielding the best average results of the object-level DIS metric for each method. For example, one can note that, although the FP score of mcRoSuRe in Table III is slightly better than the one for the mcDTSR, the superior mcDTSR TP score largely compensates for it. The optimal setup for each algorithm is shown in Table IV. In practice, one may expect that the proposed mcDTSR will have equal or superior performance when compared to mcRoSuRe-A. This is so because in cases with well aligned videos both algorithms tend to provide similar scores, but heavily misaligned videos will benefit from the domain transformation present in the mcDTSR algorithm.

Fig. 10 illustrates the superiority of mcDTSR relative to mcRoSuRe-A in the presence a significant geometric misalignment between the target and reference frames, practically eliminating the false-alarm regions on the final residue matrix. This is one of the many examples where all the previous methods fail (having DIS metric larger than 0.85) and the

TABLE II
Comparison results of the proposed mcDTSR method with STC-mc, DAOMC, MCBS, and ADMULT, considering all the 59 single-object videos of the VDAO-200 database.

Video #	STC-mc [27]			DAOMC [11]			MCBS [26]			ADMULT [6]			mcRoSuRe-A [20]			mcDTSR		
	TP	FP	DIS	TP	FP	DIS	TP	FP	DIS	TP	FP	DIS	TP	FP	DIS	TP	FP	DIS
1	0.37	0.42	0.76	1.00	1.00	1.00	1.00	0.10	0.10	1.00	0.63	0.63	1.00	0.00	**0.00**	1.00	1.00	1.00
2	1.00	0.04	0.04	1.00	0.00	**0.00**	1.00	0.90	0.90	1.00	0.00	**0.00**	0.96	0.17	0.17	0.74	0.00	0.26
3	0.90	0.04	0.11	1.00	0.04	0.04	1.00	0.28	0.28	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.70	0.70
4	1.00	0.03	0.03	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**
5	0.92	0.01	0.08	1.00	0.10	0.10	1.00	0.07	**0.07**	0.71	0.95	0.95	0.99	0.54	0.54	1.00	0.59	0.59
6	0.29	0.64	0.96	1.00	0.10	0.10	1.00	0.99	0.99	1.00	0.00	**0.00**	1.00	0.75	0.75	1.00	0.79	0.79
7	0.99	0.13	0.13	1.00	1.00	1.00	1.00	0.96	0.96	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**
8	0.00	0.01	1.00	1.00	0.87	0.87	0.75	0.31	0.39	0.54	0.02	0.47	1.00	0.16	**0.16**	1.00	0.17	0.17
9	0.00	1.00	1.41	0.94	1.00	1.00	0.67	0.18	0.37	0.52	0.06	0.48	0.94	0.00	0.06	0.97	0.00	**0.03**
10	0.01	0.01	0.99	1.00	0.97	0.97	0.89	0.10	**0.15**	0.69	0.00	0.31	0.99	0.76	0.76	0.97	0.73	0.73
11	0.03	0.79	1.25	0.98	0.98	0.98	0.73	0.32	0.42	0.67	1.00	1.05	0.94	0.05	**0.07**	0.93	0.00	**0.07**
12	0.20	0.07	0.81	0.94	0.48	0.48	0.87	1.00	1.01	1.00	0.22	0.22	0.92	0.00	**0.08**	0.90	0.00	0.10
13	0.00	0.50	1.12	0.86	0.71	0.72	0.84	0.00	0.16	0.64	0.19	0.40	0.98	0.00	0.02	1.00	0.00	**0.00**
14	0.08	0.05	0.92	1.00	0.74	0.74	0.92	0.01	0.08	1.00	0.15	0.15	0.99	0.00	**0.01**	0.88	0.00	0.12
15	0.00	1.00	1.41	1.00	1.00	1.00	0.89	1.00	1.01	0.59	0.04	0.42	1.00	0.23	0.23	1.00	0.03	**0.03**
16	0.00	0.08	1.00	0.77	1.00	1.02	0.00	0.00	1.00	0.00	0.00	1.00	0.91	0.12	0.15	0.89	0.08	**0.13**
17	0.06	1.00	1.37	0.96	0.46	0.46	0.80	0.12	0.23	0.62	0.30	0.48	0.94	0.04	0.07	0.96	0.01	**0.04**
18	0.00	0.09	1.00	0.75	0.99	1.02	0.43	0.00	0.57	0.00	0.23	1.03	0.54	0.00	0.46	0.56	0.00	**0.44**
19	0.00	0.03	1.00	1.00	0.67	0.67	0.89	0.00	0.11	0.54	0.15	0.48	1.00	0.03	**0.03**	0.95	0.00	0.05
20	0.36	0.50	**0.81**	0.26	1.00	1.24	0.67	1.00	1.05	0.00	0.00	1.00	0.99	0.97	0.97	0.80	0.98	1.00
21	0.00	0.68	1.21	0.97	0.62	0.62	0.95	0.61	0.61	0.97	0.72	0.72	1.00	0.37	0.37	1.00	0.04	**0.04**
22	0.00	0.07	1.00	1.00	0.90	0.90	0.92	0.05	0.09	0.68	0.75	0.81	1.00	0.02	**0.02**	1.00	0.04	0.04
23	0.00	0.83	1.30	0.93	1.00	1.00	0.97	1.00	1.00	1.00	1.00	1.00	0.93	0.76	0.76	0.72	0.07	**0.29**
24	0.58	0.93	1.02	0.00	1.00	1.41	0.00	0.73	1.24	0.00	0.00	1.00	0.69	1.00	1.05	0.12	0.00	**0.88**
25	0.00	0.02	1.00	1.00	0.90	0.90	0.58	0.00	**0.43**	0.56	0.55	1.00	0.51	0.00	0.50	0.55	0.01	0.45
26	0.00	0.06	1.00	1.00	0.54	0.54	0.87	0.05	0.14	0.64	0.01	0.70	0.99	0.07	**0.07**	1.00	0.53	0.53
27	0.26	0.34	0.82	1.00	0.72	0.72	1.00	1.00	1.00	1.00	0.10	0.36	1.00	0.41	0.41	1.00	0.02	**0.02**
28	0.01	0.01	1.00	1.00	0.89	0.89	1.00	0.00	**0.00**	1.00	0.00	**0.00**	0.64	0.27	0.45	0.25	0.00	0.75
29	0.00	0.14	1.01	0.91	0.98	0.98	0.76	0.02	**0.24**	0.68	0.01	0.32	0.43	0.81	0.99	0.00	0.00	1.00
30	0.00	0.01	1.00	1.00	0.97	0.97	0.80	0.49	0.53	0.56	0.00	0.44	1.00	0.36	**0.36**	1.00	0.53	0.53
31	0.00	0.01	1.00	1.00	0.61	**0.61**	0.87	0.80	0.81	0.61	0.55	0.67	0.95	0.81	0.81	0.95	1.00	1.00
32	0.00	0.01	1.00	1.00	0.78	0.78	0.83	0.00	0.17	0.32	0.00	0.68	1.00	0.01	**0.01**	0.99	0.01	0.01
33	0.78	0.81	**0.83**	0.83	1.00	1.01	1.00	1.00	1.00	1.00	1.00	1.00	0.96	1.00	1.00	0.92	1.00	1.00
34	0.00	0.02	1.00	1.00	0.69	0.69	0.70	0.00	0.30	0.56	0.00	0.44	0.95	0.02	0.05	0.97	0.03	**0.04**
35	0.00	0.97	1.39	0.97	0.62	0.62	0.87	0.82	0.83	0.62	0.01	0.38	0.94	0.81	0.81	0.96	0.00	**0.04**
36	0.24	1.00	1.26	0.02	1.00	1.40	1.00	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	0.95	0.06	**0.08**
37	0.43	0.18	0.59	0.99	1.00	0.96	0.93	0.00	0.07	0.93	0.00	0.07	0.99	0.00	**0.01**	0.97	0.00	0.03
38	0.00	1.00	1.41	1.00	0.99	0.99	0.71	0.05	0.30	0.44	0.00	0.56	0.96	0.00	**0.04**	0.72	0.00	0.28
39	0.09	0.04	0.91	0.91	1.00	1.00	0.84	0.93	0.94	1.00	0.25	**0.25**	0.91	1.00	1.00	0.92	1.00	1.00
40	0.56	0.44	0.92	1.00	0.95	0.95	1.00	0.56	0.56	1.00	0.14	**0.14**	0.92	0.28	0.29	1.00	1.00	1.00
41	0.00	0.78	1.27	0.64	0.99	1.05	0.88	0.87	0.87	0.87	1.00	1.01	0.96	1.00	1.00	1.00	0.04	**0.04**
42	0.00	1.00	1.41	0.96	0.96	0.96	0.88	0.91	0.91	0.49	0.00	0.51	0.96	0.00	0.04	0.99	0.00	**0.01**
43	0.00	0.08	1.00	0.72	1.00	1.04	0.14	0.00	0.86	0.00	0.00	1.00	0.93	0.15	0.16	0.93	0.11	**0.13**
44	0.00	0.19	1.02	0.96	1.00	1.00	0.73	0.14	0.31	0.63	0.00	0.37	0.92	0.43	0.43	0.95	0.00	**0.05**
45	0.15	0.92	1.25	0.01	1.00	1.41	0.82	1.00	1.02	1.00	1.00	1.00	0.71	1.00	1.04	0.37	0.41	**0.75**
46	0.00	0.43	1.09	0.93	0.97	0.97	0.95	0.79	0.79	0.99	0.14	0.14	0.92	0.01	**0.08**	0.91	0.00	0.09
47	0.01	0.20	1.01	1.00	1.00	1.00	0.93	0.00	**0.07**	0.91	0.22	0.24	0.98	0.30	0.30	0.97	0.26	0.26
48	0.00	0.01	1.00	0.96	0.97	0.97	0.72	0.16	0.32	0.42	0.00	0.58	0.96	0.03	0.05	0.98	0.00	**0.02**
49	0.00	0.04	1.00	1.00	0.99	0.99	1.00	0.06	**0.06**	0.93	0.00	0.07	1.00	0.24	0.24	1.00	0.76	0.76
50	0.00	0.02	1.00	1.00	0.77	0.77	0.86	0.14	0.20	0.18	0.89	1.21	0.95	0.01	0.05	0.97	0.02	**0.04**
51	0.01	0.86	1.31	0.97	0.92	0.92	0.85	0.66	**0.68**	1.00	1.00	1.00	0.94	0.98	0.98	0.81	1.00	1.02
52	0.00	0.68	1.21	0.40	1.00	1.17	0.63	0.79	0.87	0.84	1.00	1.01	0.73	1.00	1.04	0.74	0.55	**0.61**
53	0.06	0.82	1.25	0.79	1.00	1.02	0.69	1.00	1.05	0.88	1.00	1.01	0.84	0.09	**0.19**	0.85	1.00	1.01
54	0.00	0.20	1.02	1.00	0.51	0.51	0.84	0.01	0.16	0.50	0.00	0.50	0.94	0.01	0.06	1.00	0.02	**0.02**
55	0.39	0.75	0.96	0.86	1.00	1.01	0.59	0.32	0.52	0.49	0.00	0.51	0.76	0.44	0.50	0.71	0.00	**0.29**
56	0.52	0.45	0.65	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.38	**0.38**	0.98	1.00	1.00	0.81	1.00	1.02
57	0.36	0.09	0.65	0.96	0.92	0.92	1.00	0.67	0.67	1.00	0.21	0.21	0.99	0.01	**0.01**	0.93	1.00	1.00
58	0.00	0.05	1.00	0.97	0.80	0.80	0.62	0.00	0.38	0.18	0.00	0.82	0.83	0.00	0.17	0.88	0.00	**0.13**
59	0.00	1.00	1.41	1.00	1.00	1.00	0.73	0.79	0.83	0.53	0.00	0.47	0.44	0.00	0.56	0.61	0.01	**0.39**
Average	0.19	0.42	0.91	0.83	0.43	0.46	0.89	0.84	0.85	0.71	0.28	0.40	0.91	0.33	0.34	0.86	0.28	**0.31**

TABLE III
Average detection of proposed mcDTSR method compared to mcRoSuRe-A, STC-mc, DAOMC, and MCBS methods for all 59 single-object videos of the VDAO database. The same pixel-level metrics used in Table II have been employed.

Method	TP	FP	DIS
STC-mc	0.18	0.38	0.90
DAOMC	0.83	0.43	0.46
MCBS	0.89	0.84	0.85
mcRoSuRe-A	0.72	0.25	0.37
mcDTSR	0.84	0.27	**0.31**

TABLE IV
Average detection of proposed mcDTSR method, compared to STC-mc, DAOMC, MCBS, and mcRoSuRe-A methods for all 59 single-object videos of the VDAO database using frame-level metrics.

Method	TP$_{\text{fl}}$	FP$_{\text{fl}}$	DIS$_{\text{fl}}$
STC-mc	0.48	0.41	0.66
DAOMC	0.89	0.46	0.47
MCBS	0.99	0.98	0.98
mcRoSuRe-A	0.76	0.24	0.34
mcDTSR	0.88	0.26	**0.29**

proposed method excels, having a DIS metric of only 0.03.

Since the mcDTSR is based on the same sparse representation algorithm present in mcRoSuRe-A, the mcDTSR exhibits lower FP rates, in general, due to its intrinsic compensation of the camera trepidations. However, there are some situations in Table II where the mcDTSR presents more false alarms

TABLE V
CHOSEN SETUP FOR METHODS MCROSURE-A AND MCDTSR. "OPEN SIZE" AND "CLOSE SIZE" REFER TO THE DIAMETERS OF THE CORRESPONDING STRUCTURING ELEMENTS.

Method	Open Size	Close Size	Binarization Threshold	Vote Size
mcRoSuRe-A	4	1	0.20	3
mcDTSR	1	4	0.28	3

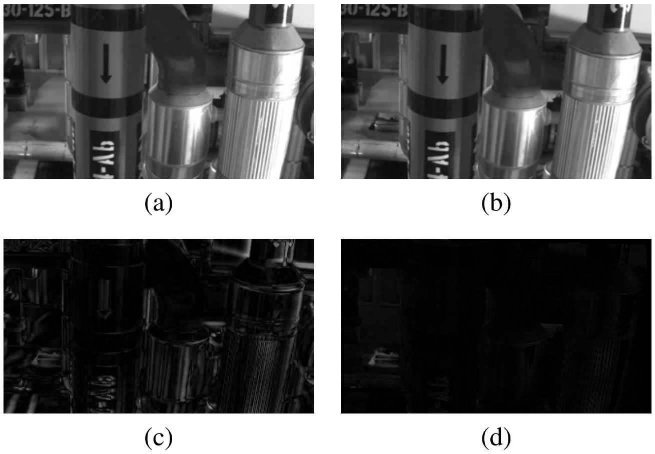


Fig. 10. Comparison of mcRoSuRe and mcDTSR residues for frame 150 of video 41 in VDAO-200 database: (a) Reference frame; (b) Target frame; (c) mcRoSuRe-A residue $|E|$; (d) mcDTSR residue $|E'|$. In this case, the mcDTSR method compensates for frame misalignment removing most of false alarms regions.

than mcRoSuRe. Having a close look at some of these cases, it is possible to see that mcDTSR captures not only the presence of abandoned objects, but also indirect visual artifacts caused by them, such as shadows and reflections, which were not considered in the ground-truth annotation for the VDAO database. This is illustrated in Fig. 11, which shows how the mcDTSR captures the shadow that the box casts in the lower pipe (Fig. 11(d)) yielding a false alarm region, which is ignored by mcRoSuRe-A method (Fig. 11(c)). Indeed, in most practical applications, this behavior, besides not being an issue, is even desirable. This is so because the goal is to find abandoned objects or anomalies, and therefore the detection of indirect artifacts caused by them is useful.

It is important to point out that the mcDTSR algorithm performs a sequence of convex optimizations in order to correct the geometric differences between the reference and target domains. In practice, a more misaligned case tends to take more steps to yield the correct alignment, and consequently, requires more processing time. Although real-time performance was not in the scope of this work, it is important for real-world applicability. One form to address this issue is to develop an accelerated version of mcDTSR along the same lines that mcRoSuRe-A [20], an accelerated version of mcRoSuRe, has been developed. Further acceleration of mcDTSR can be provided by taking advantage of the expected sparsity of the W matrix, which can reduce significantly the amount of computation in the optimization loops.

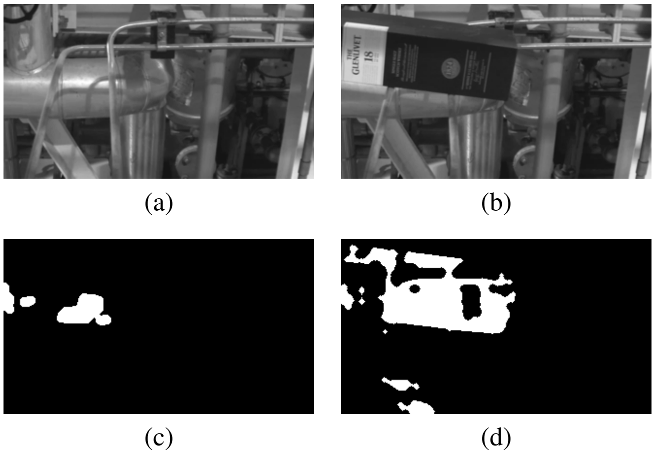


Fig. 11. Comparison of mcRoSuRe and mcDTSR results for frame 1 of video 1 in VDAO-200 database: (a) Reference frame; (b) Target frame; (c) mcRoSuRe-A detection mask; (d) mcDTSR detection mask. In this case, the shadow cast by the box, which does not have a bounding-box ground-truth counterpart, is ignored by the mcRoSuRe-A method but is successfully detected by the mcDTSR algorithm.

VI. CONCLUSION

A new algorithm was described for anomaly detection in video sequences acquired from moving cameras. The proposed system is based on the low-rank/sparse representation of target videos using a corresponding similar decomposition performed on an anomaly-free reference video. Both video representations are performed in a geometrically-transformed domain in order to compensate possible camera trepidations along its natural path. An iterative two-stage optimization procedure is employed to implement the modified optimization problem: the inner loop estimates the best geometric transformation, whereas the outer loop, given the current transformation estimate, determines the best matrix factorization. This provides a better registration between reference and target videos, reducing the amount of false alarms in the subsequent detection stage, which becomes robust to camera trepidations. Results obtained on a large database for abandoned object detection indicate superior performance of the proposed system against several state-of-the-art alternatives.

The proposed method expands the capabilities of low-rank sparse representation methods, such as mcRoSuRe-A, by incorporating, in a simple and elegant way, domain transformations that enable such methods to find more precise correspondences between different parts of the data matrix. The final algorithm has proven to be a powerful tool, in a way that it is able to outperform state-of-the-art methods in the detection of anomalies in videos acquired with a moving camera, as the results of the extensive experiments in a very challenging dataset show. The better alignment of the frames from reference and target videos allows our method to present improved true positive detection while having very few false detections.

The optimization proposed here takes inspiration on other well established domain transformation techniques, but goes further using convexification tools that reach faster convergence, which allows the method to operate in challenging scenarios.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2940686, IEEE Transactions on Image Processing

Although other recent works have proposed the use of domain transformations to improve background subtraction methods, none of the previous algorithms was able to perform in the challenging scenario considered here. This opens a path for new applications in trending areas such as video stabilization and anomaly detection with freely moving cameras, that currently lack simple tools that can be incorporated into the optimization algorithm to handle large misalignment between frames.

REFERENCES

[1] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 294–307, Mar. 2005.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computer Survey* vol. 41, no. 3, art. 15, July 2009.

[3] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "ChangeDetection.NET: A new change detection benchmark dataset," *Proc. IEEE Workshop on Change Detection*, Providence, USA, June 2012.

[4] Y. Tomioka, A. Takara, and H. Kitazawa, "Generation of an optimum patrol course for mobile surveillance camera," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, no. 22, pp. 216–224, Feb. 2012.

[5] G. N. Purohit, G. Pratishtha, and D. Amrita, "Crime prevention through alternate route finding in traffic surveillance using CCTV cameras," *Int. Journal of Engineering and Advanced Technology*, vol. 5, no. 2, pp. 414–418, 2013.

[6] G. Carvalho, L. A. Thomaz, A. F. da Silva, E. A. B. da Silva, and S. L. Netto, "Anomaly detection with a moving camera using multiscale video analysis," *Multidimensional Systems and Signal Processing*, pp.1–32, Feb. 2018.

[7] 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," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Brisbane, Australia, pp. 1295–1299, Apr. 2015.

[8] 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," *Proc. IEEE International Conference on Image Processing*, Quebec City, Canada, vol. 2, pp. 1980–1984, Sept. 2015.

[9] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras," *Proc. IEEE International Conference on Computer Vision*, pp. 1219–1225, 2009.

[10] O. M. Sincan, V. B. Ajabshir, H. Y. Keles, and S. Tosun, "Moving object detection by a mounted moving camera," *Proc. IEEE International Conference on Computer as a Tool*, pp. 1–6, Sept. 2015.

[11] H. Kong, J.-Y. Audibert, and J. Ponce, "Detecting abandoned objects with a moving camera," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2201–2210, Aug. 2010.

[12] X. Cui, J. Huang, S. Zhang, and D. N. Metaxas, "Background subtraction using low rank and group sparsity constraints," *Proc. European Conference on Computer Vision*, Part I, LNCS 7572, pp. 612–625, Oct. 2012.

[13] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment via sparse and low-rank decomposition for linearly correlated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, July 2010.

[14] E. J. Candès, Y. M. X. Li, and J. Wright, "Robust principal component analysis?" *Journal of ACM*, vol. 58, no. 3, pp. 1–37, May 2011.

[15] A. Y. Yang, Z. Zhou, A. G. Balasubramanian, S. S. Sastry, and Y. Ma, "Fast l_1-minimization algorithms for robust face recognition," *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 3234–3246, Aug. 2013.

[16] M. Soltanolkotabi and E. J. Candès, "A geometric analysis of subspace clustering with outliers," *The Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238, 2012.

[17] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[18] X. Bian and H. Krim, "Bi-sparsity pursuit for robust subspace recovery," *Proc. IEEE International Conference on Image Processing*, Quebec, Canada, Sept. 2015, pp. 3535–3539.

[19] X. Bian and H. Krim, "Robust subspace recovery via bi-sparsity pursuit," arXiv:1403.8067v2, Apr. 2014.

[20] L. A. Thomaz, E. Jardim, A. F. da Silva, E. A. B. da Silva, S. L. Netto, and H. Krim, "Anomaly detection in moving-camera video sequences using principal subspace analysis", *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 3, pp. 1003–1015, Mar. 2018.

[21] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma, "Toward a practical face recognition system: Robust alignment and illumination by sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 372–386, Feb. 2012.

[22] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma. "TILT: Transform invariant low-rank textures," *International Journal on Computer Vision*, vol. 99 no. 1, pp. 1–24, 2011.

[23] J. He, D. Zhang, L. Balzano, and T. Tao, "Iterative Grassmannian optimization for robust image alignment," *Image and Vision Computing*, vol. 32, no. 10, pp. 800–813, 2014.

[24] H. Yong, D. Meng, W. Zuo, and L. Zhang, "Robust online matrix factorization for dynamic background subtraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 7, pp. 1726–1740, 2018.

[25] Y. Wu, B. Shen, and H. Ling, "Online robust image alignment via iterative convex optimization," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Providence, USA, pp. 1808–1814, June 2012.

[26] H. Mukojima, D. Deguchi, Y. Kawanishi, I. Ide, H. Murase, M. Ukai, N. Nagamine, and R. Nakasone, "Moving camera background-subtraction for obstacle detection on railway tracks," *Proc. IEEE International Conference on Image Processing*, Phoenix, USA, pp. 3967–3971, Sept. 2016.

[27] 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", *Multidimensional Systems and Signal Processing*, vol. 24, no. 3, pp. 1025-1054, July 2018.

[28] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[29] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," *Advances in Neural Information Processing Systems*, vol. 24, pp. 612–620, 2011.

[30] A. F. da Silva, L. A. Thomaz, G. Carvalho, M. T. Nakahata, E. Jardim, J. F. L. de Oliveira, E. A. B. da Silva, S. L. Netto, G. Freitas, and R. R. Costa, "An annotated video database for abandoned-object detection in a cluttered environment," *Proc. International Telecommunications Symposium*, Sao Paulo, Brazil, Aug. 2014.

[31] C. Cuevas, R. Martínez, and N. García, "Detection of stationary foreground objects: A survey," *Computer Vision and Image Understanding*, vol. 152, pp. 41–57, November 2016.

[32] S. Dubuisson and C. Gonzales, "A survey of datasets for visual tracking," *Machine Vision and Applications*, vol. 27, no. 1, pp. 23–52, January 2016.

[33] C. Cuevas, E. M. Yáñez, and N. García, "Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA," *Computer Vision and Image Understanding*, vol. 152, pp. 103–117, November 2016.

[34] M. Camplani, L. Maddalena, G. M. Alcover, A. Petrosino, and L. Salgado, "A Benchmarking Framework for Background Subtraction in RGBD Videos," *New Trends in Image Analysis and Processing – International Conference on Image Analysis and Processing*, Catania, Italy, pp. 219–229, December 2017.

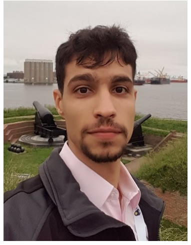
[35] S. Javed, A. Mahmood, T. Bouwmans, and S. Jung, "Spatiotemporal Low-rank Modeling for Complex Scene Background Initialization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 6, pp. 1315–1329, June 2018.

[36] S. Javed, A. Mahmood, S. Al-Maadeed, T. Bouwmans, and S. K. Jung, "Moving Object Detection in Complex Scene Using Spatiotemporal Structured-Sparse RPCA," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 1007–1022, February 2019.

[37] VDAO: Video database of abandoned objects in a cluttered industrial environment [online]. Available at:
http://www.smt.ufrj.br/%7Etvdigital/database/objects

[38] VDAO-200: 200-frame excerpts from VDAO database [online]. Available at: http://www02.smt.ufrj.br/%7Etvdigital/database/research

Eric Jardim was born in Salvador, Bahia, Brazil. He received the B.Sc. in Computer Science from Universidade Federal da Bahia (UFBA) in 2003, M.Sc. in Mathematics from Instituto Nacional de Matemática Pura e Aplicada (IMPA) in 2010, and D.Sc in Electric Engineering from Universidade Federal do Rio de Janeiro (UFRJ). Working since 2003 at Petróleo Brasileiro S.A. (PETROBRAS) with scientific computing, visualization and robotics. His research interests are in computer vision, machine learning and non-photorealistic rendering.



Lucas A. Thomaz (S'14, M'19) 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. and D.Sc. degrees in electrical engineering from COPPE/UFRJ in 2015 and 2018, respectively.

From 2017 to 2018, he was a Visiting Researcher Scholar with North Carolina State University.

Since 2019 he is a post-doc in the Instituto de Telecomunicações, Portugal. He is also involved in the JPEG standardization activities.

His research interests include the areas of computer vision, digital signal processing, video, image processing, and image and video codification.

Eduardo A. B. da Silva (M'95, SM'05) 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 is a professor of Universidade Federal do Rio de Janeiro since 1989. 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 published more than 70 papers in international journals. His research interests lie in the fields of signal and image processing, signal compression, digital TV, 3D videos, computer vision, light fields and machine learning, together with its applications to telecommunications and the oil and gas industry. He is co-editor of the future standard ISO/IEC 21794-2, JPEG Pleno Plenoptic image coding system. Prof. Da Silva is a Senior Member of the Brazilian Telecommunications Society (SbrT) and of the IEEE.

Sergio L. Netto (SM'04) was born in Rio de Janeiro, Brazil. He received the B.Sc. (cum laude) degree from the Federal University of 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, adaptive filtering, speech processing, information theory, computer vision, and machine learning.