Foreground Segmentation with Tree-Structured Sparse RPCA

Salehe Erfanian Ebadi, *Student Member, IEEE,* and Ebroul Izquierdo, *Senior Member, IEEE*

Abstract—Background subtraction is a fundamental video analysis technique that consists of creation of a background model that allows distinguishing foreground pixels. We present a new method in which the image sequence is assumed to be made up of the sum of a low-rank background matrix and a dynamic tree-structured sparse matrix. The decomposition task is then solved using our approximated Robust Principal Component Analysis (ARPCA) method which is an extension to the RPCA that can handle camera motion and noise. Our model dynamically estimates the support of the foreground regions via a superpixel generation step, so that spatial coherence can be imposed on these regions. Unlike conventional smoothness constraints such as MRF, our method is able to obtain crisp and meaningful foreground regions, and in general, handles large dynamic background motion better. To reduce the dimensionality and the curse of scale that is persistent in the RPCA-based methods, we model the background via Column Subset Selection Problem, that reduces the order of complexity and hence decreases computation time. Comprehensive evaluation on four benchmark datasets demonstrate the effectiveness of our method in outperforming state-of-the-art alternatives.

Index Terms—Approximated RPCA, structured-sparse, moving camera, dynamic background, cohesive foreground segmentation.

1 Introduction

Background subtraction can be defined as segmentation of a video sequence into the foreground and the background. It is typically used as a pre-processing step for higher level problems, such as automated surveillance, action recognition, and intelligent environments. Background subtraction poses a number of challenges in realistic environments, such as, presence of noise, illumination changes, background motions or dynamicity, camouflage, moved object, camera motion, and foreground aperture. To address these challenges, a number of considerations in designing a background model, as well as modeling the behavior of foreground objects must be made; in complex applications this is still an open problem.

Here the *noise* is modeled by the residual error of the approximation of the original data by the background plus foreground. *Illumination changes* are handled to some extent via a robust background model that is capable of adapting itself to global variations of luminance. On account of *dynamic* nature of the background, both the background model and the foreground classification mechanisms must be able to correctly classify a range of pixels. *Camouflage* is when a foreground object due to its similarity to the background persists absorbing into the background. Noting this challenge, there is a need for two semantic foreground layers, one containing genuine foreground regions, and the other ambiguous and noise-like pixels. Then, the amount to lean onto which layer for detecting foreground objects must be adaptively controlled by a robust mechanism in the model. On the other hand, a desirable background model must be able to learn a variety of modes from the video feed, such that it handles variations in the background, *moved objects*, and noise without compromising its ability to detect camouflaged regions.

Denote the set of video frames as $\{\mathbf{I}_k\}_{k=1}^{n}$ for n frames. $\mathbf{I}_k$ contains frames of a video sequence, with each image being concatenated as a column vector. We can put all the frames of a video together to form a large matrix $A = [\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_n] \in \mathbb{R}^{m \times n}$. Given the matrix A, RPCA [1] solves the matrix decomposition problem

$$\min_{L,S} \|L\|_* + \lambda \|S\|_1 \quad s.t. \quad A = L + S, \tag{1}$$

as a surrogate for the actual problem

$$\min_{L,S} rank(L) + \lambda \|S\|_0 \quad s.t. \quad A = L + S, \tag{2}$$

where L is the low-rank component corresponding to the background and S is the sparse component containing the foreground part. To overcome some inherent limitations of RPCA for background subtraction and foreground segmentation, we propose to an approximated form of the *Robust Principal Component Analysis* (RPCA) method. We are interested in the case where we can decompose the matrix A into three components, namely a low-rank part L, a sparse component S, and a residual noise part E. That is, the target is to decompose A as

$$A = L + S + E, \tag{3}$$

where E is the residual error of the approximation of A by $L + S$, that attempts to capture noise and ambiguous pixels. The decomposition above can be solved via our approximated RPCA, that we will introduce later. Observe that we expect L to be a genuine low-rank matrix, thus $rank(L) \ll rank(A)$. Moreover, by decomposing all the extra noise that contaminates the background, and storing it into E we are able to reduce the rank of the matrix L beyond what (1) is capable of. As a consequence of our decomposition strategy, the L in (3) is much more well-suited for background subtraction applications, or in general where lower dimensional models are more desirable.

Despite the promising effects of using a low-rank approximation for obtaining the background model, a sparse constraint for foreground objects, can be far too generic. In addition, processing per-pixel basis from the foreground, is not only time-consuming, but also can dramatically affect foreground region detection, if region cohesion and contiguity is not considered in the model. The foreground regions are spatially coherent clusters. Thus, we prefer to detect contiguous regions of various sizes in the matrix representing the foreground. With this objective in mind, we propose structured-sparsity inducing norms that are effective in the context of a novel dynamic group structure, by which the natural structure of foreground objects in the sparse matrix is preserved. The dynamicity of group structures is either controlled via a patch-based group selection algorithm, or derived from the natural shape of objects in the scene – by selecting clusters of pixels via the SLIC superpixels [2], and dynamically refining the size of these clusters in an iterative process. This is effective in reducing the *foreground aperture* problem with rigorous experimental evaluations.

The matrix A can become humongous when processing large or long videos. To alleviate the dimensionality and the curse of scale with an RPCA-based problem we use the *Column Subset Selection Problem* (CSSP) [3] that selects a handful of

• *S. Erfanian Ebadi and E. Izquierdo are with the department of Electronic Engineering and Computer Sceinece, Queen Mary University of London, Mile End Road, London E1 4NS, UK.E-mail: {s.erfanianebadi, e.izquierdo}@qmul.ac.uk*

Manuscript received June 2016; revised August 2017

the most representative and important columns of a matrix. Assuming that we have a long video of a scene at our disposal with hundreds or even thousands of frames, only a handful of these frames determine a model of the background; the rest will either contaminate the background or will be redundant to process. To this end, we propose to model the background of the sequence using a low-rank approximation from the output of the CSSP. Not only does this algorithm reduce the complexity and the computation time, but also alleviates the *bootstrapping* challenge, making it possible to still be able to obtain a robust model of the background without needing to observe a clean, foreground-absent frame.

In a nutshell contributions of this paper are: low-rank approximation of the background to accommodate small scene and illumination changes to some extent; inducing structured-sparsity in a novel group structure, namely a dynamic block structure and a dynamic superpixel structure; insensitivity to foreground object size, as a result of using within-patch normalization; assumption of a noise part in decomposition for reducing false positive pixels (false alarms); and a dimensionality reduction for RPCA problem via the *Column Subset Selection Problem* that alleviates *bootstrapping*, and reduces computational complexity and cost, and an analysis of the efficacy of this method. Finally, an exhaustive evaluation using four datasets [4], [5], [6], [7], demonstrating top performance in comparison with the state-of-the-art alternatives is presented.

2 Related Work

In the recent years, global models such as principal component analysis (PCA) [8], [9], [10] have gained popularity due to their simple implementation and effectiveness in camera shake. They attempt to model the background as a low-dimensional subspace of the vectorized input, with the foreground identified as outliers. In practice such approaches have struggled, due to high computational requirements and limited capability to deal with many common problems, e.g., camouflage. Recent variants have resolved part of these issues, notably [11] proposed a non-SVD based fast solution. However, still no spatial distribution of outliers were considered. In an effort to incorporate such prior an MRF-based solution [12] has been proposed. But the result of imposing such smoothness constraint is that the foreground regions tend to be over-smoothed; as an example, the details in the silhouette of hands and legs of a moving person is sacrificed in favor of a more compact blob.

Our idea is established in the so-called structured-sparsity or group-sparsity measures to incorporate the spatial prior. Structural information about nonzero patterns of variables have been developed and used in sparse signal recovery, and many approaches have been applied to these problems successfully, such as [13], [14]. However, related methods [15] typically assume that the block structure and its location is known or will suffer in *regularization* or *bootstrapping*. To lift up some difficulties [16] instead detects the block size and location by iteratively alternating between updating the block structure of the dictionary and updating the dictionary atoms to better fit the data. Nevertheless, both the number of blocks and the maximal block size are assumed to be known. In [17] and [18] the sparsity structure is estimated automatically. Parameter tuning is required in [13] to control the balance between the sparsity prior and the group clustering prior for different cases. These methods also need a clean background to train backgrounds for sequences. A two-pass RPCA framework was used in [15], where the first pass determines a saliency map generation that corresponds to locations of the outliers, and then the second pass uses pre-defined salient blocks in the image, to favor spatially contiguous outliers. In another effort [19], a group sparse structure was used, in which overlapping pre-defined groups of pixels in a region of an image are used in conjunction with a maximum norm regularization to take into account the spatial connection of foreground regions. In a recent work [20] a superpixel-based max-norm matrix decomposition approach has been proposed, in which homogeneous static or dynamic regions of image are classified as a graph partitioning problem, via Generalized Fused Lasso (GFL). In contrast, our method does not assume a prior size or location or structure for sparsity, and dynamically updates these to best fit the natural object shape in the scene, without a separate training phase.

3 Approximated RPCA with Tree-Structured Sparsity

As discussed in the introduction, our proposed approach is based on an approximated RPCA process, that takes advantage of natural structure of objects in the scene. In our model a series of structured-sparsity inducing norms are defined which act in a tree structure that is a representation of the scene components. The approximated decomposition problem stated in (3) can be solved by minimizing the decomposition error

$$\min_{L,S} \|A - L - S\|_F^2 + \lambda\|S\|_1 \quad s.t. \quad rank(L) \leq r \ll rank(A), \quad (4)$$

where $\|\cdot\|_F$ is the Frobenius norm, defined as $\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$ where a_{ij} are the elements in A. In the Frobenius norm, the set of feasible solutions is restricted to matrices L that have a rank smaller than or equal to r. It means that if r is much smaller than $min(m,n)$, the solution for L is necessarily a low-rank matrix. λ is a tuning parameter set at a value that helps recovering all genuine foreground regions. We find that using $\lambda = 3/\sqrt{max(m,n)}$ (where $m \times n$ is the dimensions of A) is adequate to identify all foreground regions in our test data. The choice of λ is justified by observations in our experiments, where λ controls a good trade-off between the sparsity of $S+E$ and structured-sparsity of S. The matrix E contains the residual error (noise) of the approximation of A by $L+S$.

3.1 Modeling with Structured-Sparsity Inducing Norms

We can exploit the natural shape of the objects in the scene to best describe the location and distribution of foreground regions; as such, we employ structured-sparsity inducing norms in the context of tree-structured groups. Also, we take into account the global background motion induced by camera motion in our model. Suppose A is an observed matrix that is not in register with the training images $\mathbf{I}_k$. To recover well-aligned images $A' = A \circ \tau$ such that they can be readily used for robust background subtraction we propose to solve the following optimization problem to seek the correct transformation τ (e.g., 2D affine transformation for correcting misalignment, or 2D projective transformation for handling some perspective change), low-rank component L, and sparse part S

$$\min_{rank(L)\leq r, S, \tau} \|A \circ \tau - L - S\|_F^2 + \lambda\|S\|_{2,1}, \quad (5)$$

where $\ell_{2,1}$-norm is a group sparsity inducing norm. Motivated by recent advances in structured sparsity [21], in this work,

we consider a tree-structured sparsity-inducing norm, that involves a hierarchical partition of the m variables in S into groups. The leaf nodes in the tree are defined to be singleton groups corresponding to individual pixels, and internal nodes/groups correspond to local patches of varying size. Thus each parent node contains a hierarchy of child nodes that are spatially adjacent to each other and constitute a local part in the sparse image S. Also when a parent node goes to zero all its descendants in the tree must go to zero. Consequently, the nonzero or support patterns are formed by removing those nodes forced to zero. This is exactly the desired effect of structured sparse patterns. We can represent a scene using a tree structure by subdivision. In such a tree structure each child node is a subset of its parent node and the nodes of the same depth level do not overlap. Denote $\mathcal{G}$ as a set of groups from the power set of the index set $\{1,\dots,m\}$, with each group $G \in \mathcal{G}$ containing a subset of these indices. The aforementioned tree-structured groups used in this paper are formally defined as follows: A set of groups $\mathcal{G}$ is said to be *tree-structured* in $\{1,\dots,m\}$ if $\mathcal{G} = \{\dots, G_1^i, G_2^i, \dots, G_{b_i}^i, \dots\}$ where $i = 0, 1, 2, \dots, d$, d is the depth of the tree, $b_0 = 1$ and $G_1^0 = \{1, 2, \dots, m\}$, $b_d = m$ and correspondingly $\{G_j^d\}_{j=1}^m$ are singleton groups. Let G_j^i be the parent node of a node $G_{j'}^{i+1}$ in the tree, we have $G_{j'}^{i+1} \subseteq G_j^i$. We also have $G_j^i \cap G_k^i = \emptyset, \forall i = 1, \dots, d, j \neq k, 1 \leq j, k \leq b_i$. Similar group structures are also considered in [22], [23]. With the above notation, a general tree-structured sparsity-inducing norm can be written as

$$\psi(S) = \sum_{i=0}^{d} \sum_{j=1}^{b_i} w_j^i \|S_{G_j^i}\|_{2,1}, \tag{6}$$

where $S_{G_j^i}$ is a vector with entries equal to those of S for the indices in G_j^i and 0 otherwise. w_j^i are positive weights for groups G_j^i. It is chosen as $w_j^i = 1/max(A_{G_j^i})$ to overcome sensitivity of the regularization scheme to illumination variance across patches. As for optimizing τ, each frame in A is sequentially aligned to each frame $\mathbf{I}_k$ instead of the whole training set $\mathbf{I}$, mainly due to the difficulty of optimization associated with the latter case, as discussed in [24]. Thus, the objective function in the optimization program is modified to the following

$$\min_{rank(L)\leq r, S, \tau} \|A \circ \tau - L - S\|_F^2 + \lambda \sum_{i=0}^{d} \sum_{j=1}^{b_i} w_j^i \|S_{G_j^i}\|_{2,1} \tag{7}$$

The problem (7) is a difficult, nonconvex optimization problem. Fortunately, we can find a good initialization by pre-aligning all frames in the sequences to the middle frame, before the main loops of minimization. The pre-alignment is done by the robust multiresolution method proposed in [25], [26]. This practice is successful in most cases given that a drastic scene change does not occur in the sequence. We can then solve (7) by repeatedly linearizing about the current estimate of τ, and seeking a deformation step $\Delta\tau$. In other words, at each iteration, we update τ by a small increment $\Delta\tau$ and linearize $A \circ \tau$ as $A \circ \tau + J\Delta\tau$, where J denotes the Jacobian matrix $J = \frac{\partial A}{\partial \tau}$. Thus, τ can be updated via the following minimization

$$\tau^t \leftarrow \tau + \arg\min_{\Delta\tau} \|A \circ \tau - L^{t-1} - S^{t-1} + J\Delta\tau\|_F^2 \tag{8}$$

The minimization over $\Delta\tau$ in (8) is a weighted least-squares problem that has a closed-form solution [26]. In practice, the update of τ for each frame can be done separately since the transformation is applied on each image individually. Thus the update of τ is efficient. We then proceed by minimizing alternatively the function for two parameters L and S one at a time until convergence

$$L^t = \arg\min_{rank(L)\leq r} \|A \circ \tau^t - L - S^{t-1}\|_F^2 \tag{9}$$

$$S^t = \arg\min_{S} \|A \circ \tau^t - L^t - S\|_F^2 + \lambda \sum_{i=0}^{d} \sum_{j=1}^{b_i} w_j^i \|S_{G_j^i}\|_{2,1} \tag{10}$$

Both these subproblems have nonconvex constraints. Their global solutions L^t and S^t exist. In particular, the two subproblems can be solved by updating L^t via singular value hard thresholding of $A - S^{t-1}$ [27], and updating S^t via our structured-sparsity inducing norms with a soft-thresholding or shrinkage operator for scalars with λ. The penalty term in (10) assures the structured-sparsity of S w.r.t. the defined tree-structured groups. The thresholding operator is defined as

$$\mathcal{P}_\lambda(x) = \begin{cases} \text{sign}(x)\max(|x| - \lambda), & |x| > \lambda, \\ 0, & |x| \leq \lambda. \end{cases} \tag{11}$$

3.2 Tree-Structured Groups in Meaningful Regions

There is a need for some mechanism that can take into account the natural shape and structure of objects in the scene, in the structured-sparse solution with the tree-structured group $\psi(\cdot)$. Each group must take into account connected components belonging to a semantically or texturally connected region. For example, a region of pixels with the same color and texture belonging to part of an object (a wheel of a car) must be assigned to a single group. The structured-sparse inducing framework defined in the previous section can then be used within the group class to decide whether it belongs to foreground or must be classified as background.

A trend in recent literature has been shifting towards a very common approach in video coding technology, where the test image is divided into square-shaped regions of pixels called blocks, with pre-determined sizes. To achieve even more sophisticated sectioning, each block can be further divided into smaller blocks each time halving the size of the block. This can be done until a block of size 1 (a single pixel) is reached; this is called the quad-tree decomposition. This approach is not very complex and can be implemented with low order of computation in the framework we described in the previous sections. In this example a region of 8×8 pixels is chosen as a group. If there are no elements with large magnitude in this region, the sparsity-inducing norms will classify the whole region to background; otherwise it is divided into 4 smaller regions of 4×4 pixels. Similarly, each of the smaller regions are put to the test of sparsity-inducing norms, and the regions belonging to background are left-off, while the regions hinting foreground elements are divided into 4 smaller regions once again. This is done until a singleton group (a single pixel) is reached. We call this procedure *induction*, *division*, and *discarding*. There are two immediate benefits from defining such a block structure: firstly, the amount of computation needed for classification is lowered, as classifying larger regions to background is much faster compared to single pixel assignment, while for blocks containing foreground objects the subdivisions will allow more meticulous investigation in these regions. Large region classification can be safely done in our model; this is the direct impact of our sparsity-inducing norms definition, since despite other RPCA-based methods our algorithm is not sensitive to the size

of the region in question. Secondly, the recursive division of regions down to one pixel will result in a very crisp and well-defined foreground segmentation. We refer to this approach in this paper as *DBSS* model, short-hand for *Dynamic Block Structured Sparse*. Depth of each tree in this model is set to $d = 3$ and $m = 64$, therefore $\mathcal{G} = \{\dots, G_1^i, G_2^i, \dots, G_{b_i}^i, \dots\}$ where $i = \{0, 1, 2, 3\}$, $b_0 = 1$ and $G_1^0 = \{1, 2, \dots, 64\}$, $b_d = 64$ and correspondingly $\{G_j^d\}_{j=1}^{64}$ are singleton groups. The general tree-structured sparsity-inducing norm becomes

$$\psi(S) = \sum_{i=0}^{3} \sum_{j=1}^{b_i} w_j^i \|S_{G_j^i}\|_{2,1} \tag{12}$$

As mentioned before, DBSS bears two limitations that the size and location of the blocks need to be set in advance, and it is hard to see how each block is adapting its shape to the natural structure of objects in the scene. Motivated by these limitations, we propose a new group structure, in which the sparse part derives its structure from the natural object structure in the scene. In a test image, the scene can be classified into multiple *superpixels*. Recent advances in image segmentation have made a plethora of superpixel algorithms becoming available, that promise state-of-the-art ability to adhere to image boundaries, speed, memory efficiency, and segmentation performance. A good superpixel must obtain perceptually meaningful atomic regions, which can be used to replace the rigid structure of the pixel grid. Moreover, as these results will be used as a pre-processing step in our foreground detection framework, they should be fast to compute, memory efficient, and simple to use. Also, in our segmentation scenario, superpixels should both increase the speed and improve the quality of the results.

We therefore, adopt the *Simple Linear Iterative Clustering* (SLIC) algorithm based on the empirical comparison of six state-of-the-art superpixel methods [2]. SLIC adapts *k*-means clustering to generate superpixels, and is freely available[1]. By default, the only parameters of the algorithm are the desired number of approximately equally-sized superpixels, and a compactness factor controlling adherence of each superpixel region to object boundaries. It seems that for our test images, 800 superpixels are sufficient to adhere well to all object boundaries.

Once the superpixels are obtained in the pre-processing step, the same procedure for structured sparsity inducing norms is applied to groups, that are this time each superpixel region in the test image. For recursive division however we cannot follow the naïve recursive block division of DBSS. We have adapted SLIC to be able to dynamically divide each superpixel region into approximately equal-sized smaller superpixels. Each initial superpixel region is divided into 4 smaller superpixels that best adhere to object boundaries. These smaller superpixels are further divided into 4 regions, again and again. Our experiments have shown that at this depth the classification can be performed without having to perform any further divisions, as the regions are both small enough to safely discard non-foreground regions, and large enough to crisply classify all foreground objects in the scene with fine details correctly. We denote this model as *DSPSS* short for *Dynamic SuperPixel Structured Sparse*. Similarly the parameters for the tree-structured sparsity-inducing norm $\psi(s)$ are defined as follow. Depth of each tree in this model is $d = 3$ and $m = \mathcal{M}$ is dynamically decided by SLIC, since it depends on image size, and the natural shape of the objects in the scene. Therefore $\mathcal{G} = \{\dots, G_1^i, G_2^i, \dots, G_{b_i}^i, \dots\}$ where $i = \{0, 1, 2, 3\}$, $b_0 = 1$ and $G_1^0 = \{1, 2, \dots, \mathcal{M}\}$, $b_d = \mathcal{M}$ and correspondingly $\{G_j^d\}_{j=1}^{\mathcal{M}}$ are the smallest superpixel groups. A summary of DBSS and DSPSS methods is described in Algorithm 1; the operator ψ determines which algorithm is used. To initialize values for the matrices L and S in both DBSS and DSPSS we use a novel Tandem initialization method [28] that results in faster convergence of the iterative process, yields more stable results, and increases the segmentation accuracy.

1. http://ivrl.epfl.ch/research/superpixels

Algorithm 1 Pseudo-code for DBSS and DSPSS with background motion parameter estimation and Tandem initialization

1: **Input:** A, *rank*, λ, ϵ, *maxIter*
2: **Output:** S, L, E, τ
3: *Tandem initialization:* $\tau^0 = 0$, $L^0 =$ rank-r approximation of A, $S^0 = A - L^0$
4: **while** $\|A \circ \tau^t - L^t - S^t\|_F^2 / \|A\|_F^2 > \epsilon$ or $t <$ *maxIter* **do**
   1) Form the matrix $A \circ \tau$ calculating the parameters τ_i^t that infer the mapping that transforms the column vector A_i to the i-th column vector of the matrix $L^{t-1} + S^{t-1}$.
   2) Calculate $L^t = \sum_{i=1}^{rank} \sigma_i U_i V_i^T$ where $\text{svd}(A \circ \tau^t - S^{t-1}) = U \Sigma V^T$.
   3) Calculate $S^t = \mathcal{P}_\lambda(\psi(A \circ \tau^t - L^t))$ where $\mathcal{P}_\lambda(x) = \text{sign}(x) \max(|x| - \lambda, 0)$.
   4) Calculate the residual noise $E = A - L - S$.
5: **end while**

3.3 Dimensionality Reduction for Decomposition

Computational cost of RPCA methods lies mainly in the *SVD* calculation step for low-rank minimization. As the resolution of the images or the length of the video increase, RPCA becomes progressively computationally inefficient. There exist deterministic algorithms for solving the *Column Subset Selection Problem* (CSSP), that use probability distributions to find the most representative columns in a matrix [3]. The CSSP is defined as: Let $A \in \mathbb{R}^{m \times n}$ and let $c \ll n$ be a sampling parameter. Find c columns of A – denoted as $C \in \mathbb{R}^{m \times c}$ – that minimize

$$\|A - CC^\dagger A\|_F \quad or \quad \|A - CC^\dagger A\|_2, \tag{13}$$

where $C^\dagger$ denotes the Moore-Penrose pseudo-inverse. We can equivalently write $C = A\mathcal{A}$, where the *sampling matrix* is $\mathcal{A} \in \mathbb{R}^{n \times c}$. A simple but extremely successful deterministic strategy is proposed [29] which is based on sampling columns of A that correspond to the largest leverage scores ℓ_i^κ, for some $\kappa < rank(A)$. As the number of columns to be selected is not known a priori, the algorithm selects the c columns of A that correspond to the largest c leverage scores ℓ_i^κ such that their sum $\sum_{i=1}^{c} \ell_i^\kappa$ is more than an "energy" parameter θ with $c = \theta \times n$. We choose θ such that $rank(V_\kappa^T \mathcal{A}) = \kappa$; where $V_\kappa \in \mathbb{R}^{n \times \kappa}$ contains the top κ right singular vectors of the matrix $A \in \mathbb{R}^{m \times n}$ with rank $r = rank(A) \geq \kappa$. Then, The rank-κ leverage score of the i-th column of A is defined as

$$\ell_i^\kappa = \|V_\kappa(i, :)\|_2^2, \quad i = 1, 2, \dots, n, \tag{14}$$

Here, $V_\kappa(i, :)$ denotes the i-th row of V_κ. Based on the presented methodology, we perform the background modeling using the output of CSSP algorithm, where a lot of redundant information is discarded, as it does not contribute to the background model, if even worse, does not contaminate it [30].

4 Experiments and Analysis

We present qualitative and quantitative results for two algorithms proposed in this paper, DBSS and DSPSS both with tandem initialization [28] and CSSP for background modeling. All

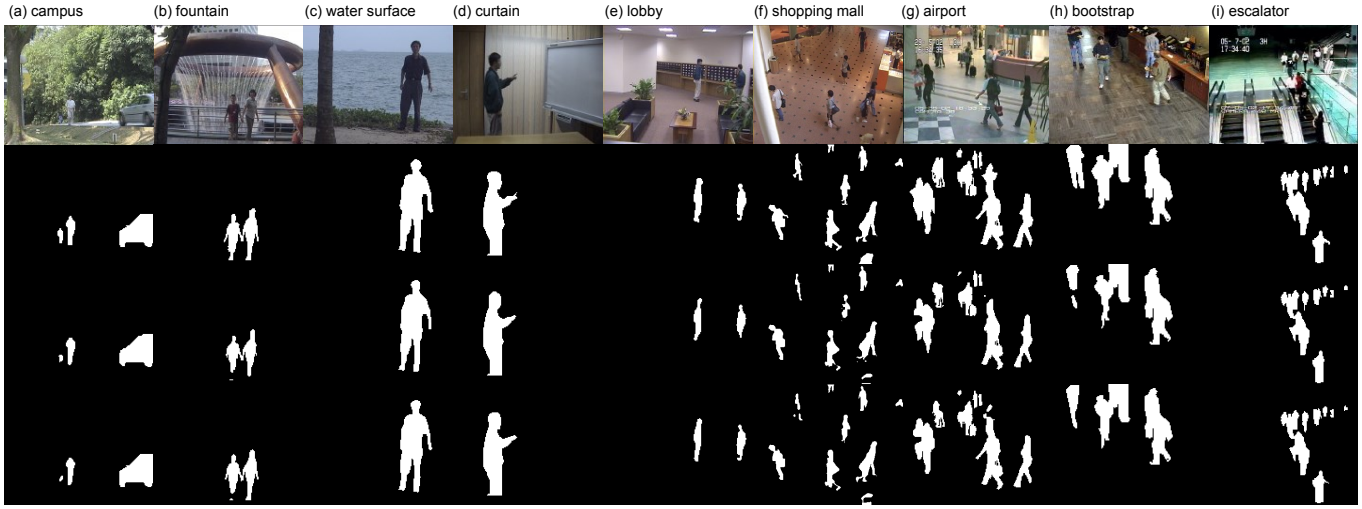

Fig. 1: *i2R* [6] results: top row is the original image, second row is the ground truth, the third row is DBSS results, and the last row is DSPSS output. We used the same frames as [20], [19], [31], [32], [33], and [34], for qualitative comparison.

TABLE 1: Description of the parameters for DBSS and DSPSS.

DBSS	λ	Tuning parameter.	$3/\sqrt{\max(m,n)}$
	d	Depth of each tree.	3
	m	# singleton groups.	64
	θ	Energy value for CSSP.	.25
DSPSS	λ	Tuning parameter.	$3/\sqrt{\max(m,n)}$
	d	Depth of each tree.	3
	$\mathcal{M}$	# singleton groups.	*Dynamic*
	k_clusters	# superpixels per image.	800
	c_factor	Compactness factor	20
	θ	Energy value for CSSP.	.25

the tests were conducted on the *temporal region of interest* of the sequences, meaning no training stage with clean background was used to obtain the background model. The algorithms are implemented in MATLAB and run on a desktop machine, using a single core on an Intel Core i7-4770 CPU and 32 GB of RAM. The average processing time for a sequence of 100 RGB frames with resolution 600×800 with image alignment and background motion estimation, and without CSSP is about 665 seconds for DBSS and 1674 seconds for DSPSS excluding the superpixel generation step. With CSSP these times decrease accordingly to 195 seconds for DBSS and 488 seconds for DSPSS, meaning that time consumption is decreased more than 3.4 times. It is worth mentioning that the amount of time required for RPCA-based methods substantially increases with the number of frames, and one would eventually run out of memory. Hence, without CSSP, the time consumption trend is non-linear and going to explode.

Four datasets are used in our experiments: *SABS* [5], *WallFlower* [4], *i2R* [6], and *Change Detection (CDnet) 2012* [7]. We perform extensive tests using these datasets comprised of a total of 49 videos, allowing us to compare our method to a large number of alternative methods. For all the tests the same set of parameters are used (reported in table 1).

4.1 CDnet 2012 Dataset

The results for these sequences can be seen in figure 2. Quantitative results can be found in table 2. In addition to this list, we have included the DP-GMM [32] and five RPCA-based methods PCP [35], DECOLOR [12], LSD-GSRPCA [19], SPGFL [20], and very recent 2-pass RPCA [15]. For LSD-GSRPCA [19] and SPGFL [20] only a fraction of the results were reported in their papers, therefore they are included where results are reported. For PCP we use our pre-alignment step for the *camera jitter* sequences and refer to it as PCP+Alignment. The online version of CDnet combines many different scoring mechanisms, and then combines them in a non-linear rank based system. Instead, we present the F-measure scores, as it is the most used metric. The F-measure is defined as the harmonic mean of the recall and precision

$$\text{recall} = \frac{tp}{tp+fn}, \quad \text{precision} = \frac{tp}{tp+fp}, \tag{15}$$

$$\text{F-measure} = 2\frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}}, \tag{16}$$

where fp is the number of false positives, tn the number of true negatives, etc. Overall, we win on average for the extremely challenging CDnet dataset both for DBSS and DSPSS. This is because our model can handle backgrounds that are complex and dynamic. This ability, in combination with the tree-structured sparsity inducing mechanisms allows it to effectively segment genuine well-outlined foreground regions.

4.2 SABS Dataset

As can be seen in the results in table 3, our DSPSS algorithm takes the first place in all the scenarios except for *light switch*. Our background model slowly adapts to changes in the scene, and this takes its toll on our method in this challenge. The DSPSS wins on average, and DBSS stands 3$^{\text{rd}}$ after DP-GMM.

4.3 i2R and WallFlower Datasets

The i2R dataset [6] and WallFlower [4] datasets share 2 videos, therefore, we report their results together. The testing procedure is similar to before. We have reported for DBSS and DSPSS results with and without parameter tuning per problem, since some methods in comparison have used tuning and some have not. The qualitative results can be seen in figure 1 and F-measure results can be seen in table 4. We achieve top performance again in all categories except for *lb* sequence, that contains abrupt lighting changes, which is compensated for slowly by our background model. Our DBSS algorithm without parameter tuning in this table achieves a modest 5$^{\text{th}}$ place as a

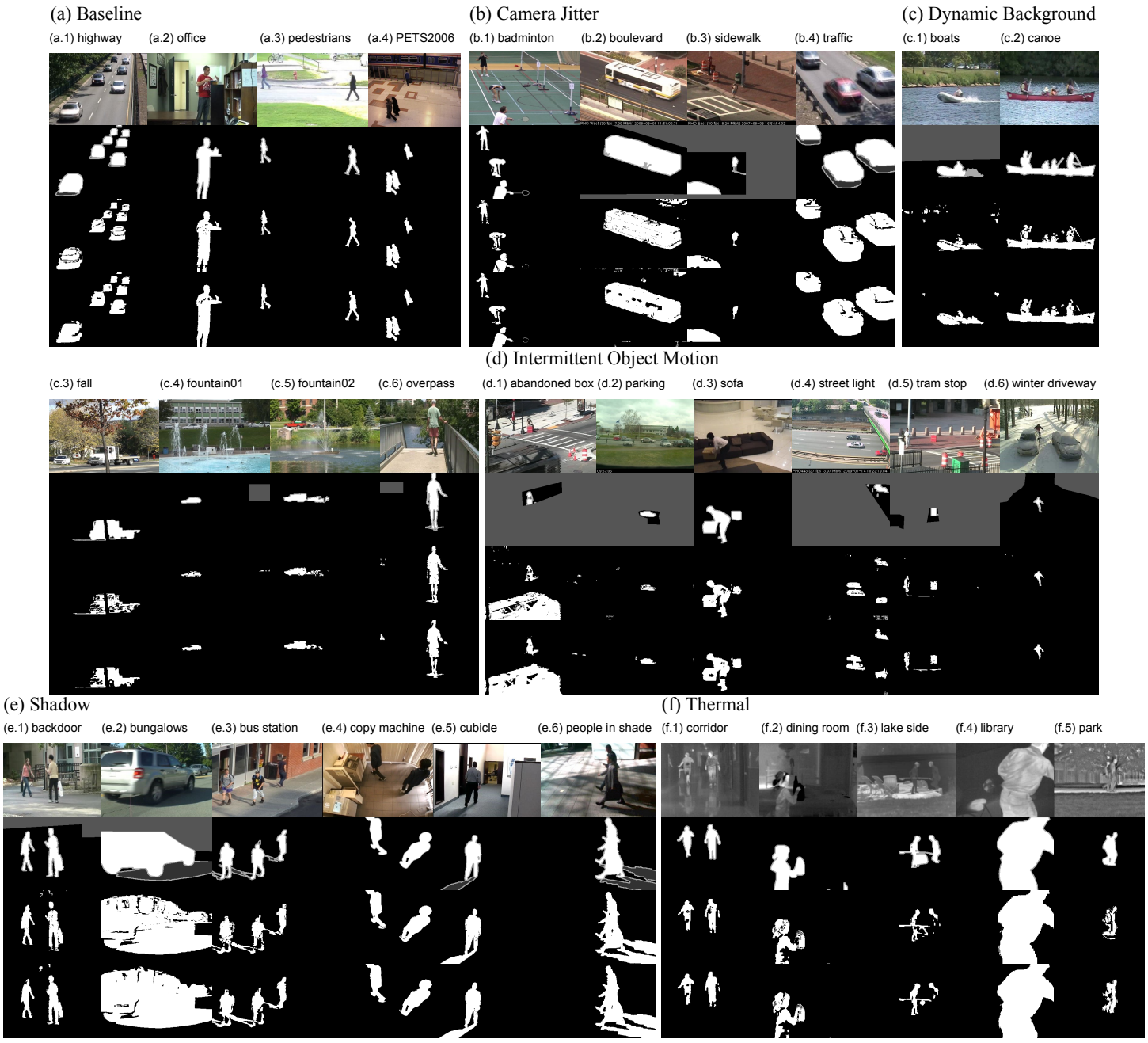

Fig. 2: *CDnet 2012* [7] results: identical layout to figure 1 with multiple rows. The ground truth includes is marked with various shades of gray – dark gray to indicate shadows, mid gray for ignored regions for evaluation, and light gray for areas ignored per frame, usually the outline of objects where foreground/background assignment is ambiguous.

result of suffering during *lb*, but the DSPSS remains at the top place regardless.

5 Conclusion

A new background subtraction method was presented and its efficacy and effectiveness was validated with extensive testing. The method is an extension to an existing model, namely RPCA, but with additional noise and motion transformation components, new sparsity-inducing norms, and group-structured sparsity constraints. Our sparsity models dynamically evolve to best describe genuine foreground objects in the scene, which gives them a significant advantage when it comes to handling dynamic backgrounds, or foreground aperture. To make the problem computationally scalable we proposed using deterministic and randomized CSSP for low-rank matrix estimation and analyzed its efficacy rigorously. Moreover, a novel tandem initialization method is used to speed up convergence and remove ghosting effects persisting in RPCA-based methods. Specifically, our model is able to learn a robust background model that can change over time, to cope with a variety of scene changes, in comparison with the existing more heuristic RPCA-based methods. It proves itself to have excellent performance in dealing with heavy noise, thanks to the approximated RPCA model where the residual error (noise) is discarded into a third matrix in the decomposition. In addition, estimation of background motion induced by a jittering

TABLE 2: *CDNet 2012* [7] dataset: F-measure results of all the categories for the most competitive methods. Table accurate as of August 2017, with results from CDnet http://changedetection.net/. The online chart keeps updating.

method	Baseline	Camera Jitter	Dynamic Background	Intermittent Motion	Shadow	Thermal	*mean*
LSD-GSRPCA [19]	.7173 (19)	-	-	-	-	-	-
SPGFL [20]	.9469 (3)	-	.8519 (5)	.6988 (7)	-	.8156 (5)	-
SGMM [36]	.8594 (18)	.7251 (13)	.6380 (18)	.5397 (16)	.7944 (14)	.6481 (18)	.7008 (17)
ViBe+ [37]	.8715 (17)	.7538 (10)	.7197 (11)	.5093 (18)	.8153 (9)	.6646 (17)	.7224 (16)
SC-SOBS [38]	.9333 (7)	.7051 (16)	.6686 (17)	.5918 (12)	.7786 (17)	.6923 (16)	.7283 (15)
PCP+Alignment [35]	.9109 (16)	.7218 (15)	.6941 (14)	.5371 (17)	.7885 (16)	.7192 (12)	.7286 (14)
PSP-MRF [39]	.9289 (8)	.7502 (11)	.6960 (13)	.5645 (14)	.7907 (15)	.6932 (15)	.7372 (13)
PBAS [40]	.9242 (13)	.7220 (14)	.6829 (15)	.5745 (13)	.8597 (6)	.7556 (9)	.7532 (12)
DECOLOR [12]	.9215 (15)	.7776 (9)	.7084 (12)	.5945 (11)	.8317 (7)	.7081 (14)	.7570 (11)
SGMM-SOD [41]	.9223 (14)	.6988 (17)	.6826 (16)	.6957 (8)	.8613 (5)	.7081 (13)	.7624 (10)
DP-GMM [32]	.9286 (11)	.7477 (12)	.8137 (7)	.5418 (15)	.8127 (10)	.8134 (6)	.7763 (9)
2-pass RPCA [15]	.9281 (12)	.8152 (6)	.7818 (10)	.6826 (9)	.8063 (13)	.7597 (8)	.7956 (8)
MBS V0 [42]	.9287 (10)	.8367 (5)	.7904 (9)	.7092 (6)	.8063 (12)	.8115 (7)	.8092 (7)
MBS [43]	.9287 (9)	.8367 (4)	.7915 (8)	.7568 (5)	.8262 (8)	.8194 (3)	.8217 (6)
SuBSENSE [44]	.9500 (2)	.8150 (7)	.8180 (6)	.6570 (10)	.8990 (3)	.8170 (4)	.8260 (5)
PAWCS [45]	.9397 (6)	.8137 (8)	.8938 (4)	.7764 (4)	.8710 (4)	.8324 (2)	.8545 (4)
CDet [46]	.9458 (4)	.8367 (3)	.8991 (3)	**.8039 (1)**	.8122 (11)	**.8337 (1)**	.8552 (3)
DBSS	.9430 (5)	**.8804 (1)**	.9005 (2)	.7837 (3)	.9107 (2)	.7195 (11)	.8563 (2)
DSPSS	**.9664 (1)**	.8662 (2)	**.9057 (1)**	.7870 (2)	**.9177 (1)**	.7328 (10)	**.8626 (1)**

TABLE 3: *SABS* [5] dataset: F-measure results for nine challenges; only the most competitive algorithms were included.

method	basic	dynamic background	bootstrap	darkening	light switch	noisy night	camouflage	no camouflage	H264, 40kbps	*mean*
Stauffer [47]	.800 (4)	.704 (6)	.642 (6)	.404 (8)	.217 (7)	.194 (7)	.802 (5)	.826 (5)	.761 (7)	.594 (8)
Maddalena [34]	.766 (6)	.715 (4)	.495 (8)	.663 (6)	.213 (8)	.263 (6)	.793 (6)	.811 (6)	.772 (6)	.610 (7)
Li 1 [48]	.766 (6)	.641 (7)	.678 (5)	.704 (4)	.316 (4)	.047 (8)	.768 (7)	.803 (7)	.773 (5)	.611 (6)
Barnich [49]	.761 (7)	.711 (5)	.685 (4)	.678 (5)	.268 (6)	.271 (5)	.741 (8)	.799 (8)	.774 (4)	.632 (5)
Zivkovic [50]	.768 (5)	.704 (6)	.632 (7)	.620 (7)	.300 (5)	.321 (4)	.820 (4)	.829 (4)	.748 (8)	.638 (4)
DP-GMM [32]	.853 (2)	.853 (2)	.796 (3)	.861 (2)	**.603 (1)**	.788 (2)	.864 (3)	.867 (3)	.827 (2)	.812 (2)
DBSS	.823 (3)	.701 (3)	.798 (2)	.850 (3)	.496 (3)	.715 (3)	.878 (2)	.890 (2)	.806 (3)	.784 (3)
DSPSS	**.867 (1)**	**.871 (1)**	**.822 (1)**	**.907 (1)**	.570 (2)	**.897 (1)**	**.894 (1)**	**.913 (1)**	**.841 (1)**	**.842 (1)**

TABLE 4: *i2R* [6] and *WallFlower* [4] dataset F-measure results. We report DBSS* and DSPSS* without parameter tuning, although the dataset allows this.

method	cam	ft	ws	mc	lb	sm	ap	br	ss	*mean*
Li 2 [6]	.1596 (11)	.0999 (14)	.0667 (14)	.1841 (14)	.1554 (14)	.5209 (14)	.1135 (14)	.3079 (14)	.1294 (14)	.1930 (14)
SSGoDec [27]	.0903 (12)	.2574 (12)	.4473 (13)	.4344 (13)	.3602 (13)	.6554 (11)	.5713 (10)	.3561 (13)	.2751 (12)	.3830 (13)
Stauffer [47]	.7570 (6)	.6854 (9)	.7948 (10)	.7580 (11)	.6519 (8)	.5363 (13)	.3335 (13)	.3838 (12)	.1388 (13)	.4842 (12)
Culibrk [33]	.5256 (8)	.4636 (11)	.7540 (11)	.7368 (12)	.6276 (11)	.5696 (12)	.3923 (12)	.4779 (11)	.4928 (11)	.5600 (11)
DECOLOR [12]	.3416 (10)	.2075 (13)	.9022 (8)	.8700 (6)	.646 (10)	.6822 (8)	.8169 (4)	.6589 (7)	.7480 (6)	.6525 (10)
Maddalena [34]	.6960 (7)	.6554 (10)	.8247 (9)	.8178 (10)	.6489 (9)	.6677 (10)	.5943 (8)	.6019 (9)	.5770 (9)	.6760 (9)
DP-GMM [32]	.7876 (4)	.7424 (8)	.9298 (5)	.8411 (8)	.6665 (7)	.6733 (9)	.5675 (11)	.6496 (8)	.5522 (10)	.7122 (8)
PCP [35]	.5226 (9)	.8650 (5)	.6082 (12)	.9014 (5)	.7245 (6)	.7785 (6)	.5879 (9)	.8322 (6)	.7374 (7)	.7286 (7)
LSD-GSRPCA [19]	.7613 (6)	.8371 (6)	.9050 (7)	.8357 (9)	.7313 (5)	.7362 (7)	.7222 (7)	.5842 (10)	.7214 (8)	.7594 (6)
SPGFL [20]	.8574 (4)	.9322 (2)	**.9856 (1)**	**.9744 (1)**	**.8840 (1)**	.8265 (4)	.7739 (5)	.8394 (5)	.8029 (5)	.8751 (4)
DBSS*	.8173 (5)	.7842 (7)	.9282 (6)	.8565 (7)	.5838 (12)	.8071 (5)	.7379 (6)	.8645 (4)	.8586 (4)	.8042 (5)
DBSS, tuned	.9277 (2)	.8808 (4)	.9535 (4)	.9093 (4)	.7563 (4)	.8950 (2)	.8343 (3)	.9196 (2)	.9377 (2)	.8904 (2)
DSPSS*	.8993 (3)	.9105 (3)	.9674 (3)	.9228 (2)	.7680 (3)	.8499 (3)	.8593 (2)	.8922 (3)	.9163 (3)	.8873 (3)
DSPSS, tuned	**.9610 (1)**	**.9575 (1)**	.9719 (2)	.9093 (3)	.8725 (2)	**.9156 (1)**	**.9098 (1)**	**.9440 (1)**	**.9561 (1)**	**.9331 (1)**

or moving camera is performed simultaneously with low-rank approximation, that results in excellent performance in videos with large camera-induced motion. Our model is however, yet another batch method, as the frames need to be stored for obtaining a background model; although we alleviated this limitation to some extent by the CSSP, further optimization is required to achieve real-time performances. This could include a learning stage followed by incremental updates as the frames arrive. Spatio-temporal constraints are also another area for future studies. Our model fails to handle some sudden illumination and lighting changes as the background model slowly adapts to these. Furthermore, our model could take advantage from a mechanism to handle shadows with more sophisticated processing.

ACKNOWLEDGMENTS

This work was supported in part by the LASIE project (http://www.lasie-project.eu/) with funding from the European Unions Seventh Framework Program for research and technological development.

REFERENCES

[1] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, Jun. 2011.
[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2274–2282, Nov 2012.
[3] D. Papailiopoulos, A. Kyrillidis, and C. Boutsidis, "Provable deterministic leverage score sampling," in *Proceedings of the 20th ACM SIGKDD*

international conference on Knowledge discovery and data mining. ACM, 2014, pp. 997–1006.

[4] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, 1999, pp. 255–261 vol.1.

[5] S. Brutzer, B. Höferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Computer Vision and Pattern Recognition (CVPR) IEEE*, 2011, pp. 1937–1944.

[6] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1459–1472, 2004.

[7] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *IEEE CVPR Change Detection workshop*, United States, Jun. 2014, p. 8 p., https://hal-univ-bourgogne.archives-ouvertes.fr/hal-01018757.

[8] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831–843, 2000.

[9] C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Foreground detection via robust low-rank matrix decomposition including spatio-temporal constraint," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 315–320.

[10] A. Sobral, T. Bouwmans, and E.-H. ZahZah, "Double-constrained RPCA based on saliency maps for foreground detection in automated maritime surveillance," in *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*. IEEE, 2015, pp. 1–6.

[11] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1568–1575.

[12] X. Zhou, C. Yang, and W. Yu, "DECOLOR: Moving object detection by detecting contiguous outliers in the low-rank representation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 597–610, 2013.

[13] J. Huang, X. Huang, and D. Metaxas, "Learning with dynamic group sparsity," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 64–71.

[14] J. Mairal, R. Jenatton, F. R. Bach, and G. R. Obozinski, "Network flow algorithms for structured sparsity," in *Advances in Neural Information Processing Systems*, 2010, pp. 1558–1566.

[15] Z. Gao, L.-F. Cheong, and Y.-X. Wang, "Block-sparse RPCA for salient motion detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 10, pp. 1975–1987, Oct 2014.

[16] L. Zelnik-Manor, K. Rosenblum, and Y. C. Eldar, "Dictionary optimization for block-sparse representations," *Signal Processing, IEEE Transactions on*, vol. 60, no. 5, pp. 2386–2395, 2012.

[17] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.

[18] C. Qiu and N. Vaswani, "ReProCS: A missing link between recursive robust PCA and recursive sparse recovery in large but correlated noise," *arXiv preprint arXiv:1106.3286*, 2011.

[19] X. Liu, G. Zhao, J. Yao, and C. Qi, "Background subtraction based on low-rank and structured sparse decomposition," 2015.

[20] S. Javed, S. Oh, A. Sobral, T. Bouwmans, and S. Jung, "Background subtraction via superpixel-based online matrix decomposition with structured foreground constraints," in *Workshop on Robust Subspace Learning and Computer Vision, ICCV 2015*, 2015.

[21] R. Jenatton, J.-Y. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2777–2824, 2011.

[22] L. Jacob, G. Obozinski, and J.-P. Vert, "Group Lasso with overlap and graph Lasso," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 433–440.

[23] S. Kim and E. P. Xing, "Tree-guided group Lasso for multi-task regression with structured sparsity," 2010.

[24] K. Jia, T.-H. Chan, and Y. Ma, "Robust and practical face recognition via structured sparsity," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 331–344.

[25] J.-M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of visual communication and image representation*, vol. 6, no. 4, pp. 348–365, 1995.

[26] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, 2012.

[27] T. Zhou and D. Tao, "GoDec: Randomized low-rank and sparse matrix decomposition in noisy case," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ser. ICML '11, L. Getoor and T. Scheffer, Eds. ACM, June 2011, pp. 33–40.

[28] S. Erfanian Ebadi and E. Izquierdo, "Foreground segmentation via dynamic tree-structured sparse RPCA," in *Computer Vision (ECCV), 2016 IEEE European Conference on*, 2016.

[29] I. T. Jolliffe, "Discarding variables in a principal component analysis. i: Artificial data," *Applied statistics*, pp. 160–173, 1972.

[30] S. Erfanian Ebadi, V. G. Ones, and E. Izquierdo, "Dynamic tree-structured sparse RPCA via column subset selection for background modeling and foreground detection," in *Image Processing (ICIP), 2016 IEEE International Conference on*, Sept 2016.

[31] B. Xin, Y. Tian, Y. Wang, and W. Gao, "Background subtraction via generalized fused Lasso foreground modeling," *arXiv preprint arXiv:1504.03707*, 2015.

[32] T. Haines and T. Xiang, "Background subtraction with dirichlet process mixture models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 4, pp. 670–683, April 2014.

[33] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht, "Neural network approach to background modeling for video object segmentation," *Neural Networks, IEEE Transactions on*, vol. 18, no. 6, pp. 1614–1627, 2007.

[34] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Transations on Image Processing*, vol. 17, no. 7, pp. 1168–1177, July 2008. [Online]. Available: http://dx.doi.org/10.1109/TIP.2008.924285

[35] Z. Zhou, X. Li, J. Wright, E. J. Candès, and Y. Ma, "Stable principal component pursuit," *CoRR*, vol. abs/1001.2363, 2010. [Online]. Available: http://arxiv.org/abs/1001.2363

[36] R. H. Evangelio, M. Pätzold, and T. Sikora, "Splitting gaussians in mixture models," in *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*. IEEE, 2012, pp. 300–305.

[37] O. Barnich and M. V. Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, pp. 1709–1724, 2011.

[38] L. Maddalena and A. Petrosino, "The SOBS algorithm: What are the limits?" in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 21–26.

[39] A. Schick, M. Bäuml, and R. Stiefelhagen, "Improving foreground segmentations with probabilistic superpixel markov random fields," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 27–31.

[40] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 38–43.

[41] R. H. Evangelio and T. Sikora, "Complementary background models for the detection of static and moving objects in crowded environments," in *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*. IEEE, 2011, pp. 71–76.

[42] H. Sajid and S.-C. S. Cheung, "Background subtraction for static & moving camera," in *Image Processing (ICIP), 2015 IEEE International Conference on*.

[43] ——, "Universal multimode background subtraction," in *Image Processing (ICIP), Submitted to 2015 IEEE International Conference on*.

[44] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *Image Processing, IEEE Transactions on*, vol. 24, no. 1, pp. 359–373, 2015.

[45] ——, "A self-adjusting approach to change detection based on background word consensus," in *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*. IEEE, 2015, pp. 990–997.

[46] Anonymous, "CDet," 2012. [Online]. Available: http://wordpress-jodoin.dmi.usherb.ca/method/146/

[47] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2. IEEE, 1999.

[48] L. Li, W. Huang, I. Y. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proceedings of the eleventh ACM international conference on Multimedia*. ACM, 2003, pp. 2–10.

[49] O. Barnich and M. Van Droogenbroeck, "ViBe: a powerful random technique to estimate the background in video sequences," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 945–948.

[50] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern recognition letters*, vol. 27, no. 7, pp. 773–780, 2006.