# Background Subtraction via Superpixel-based Online Matrix Decomposition with Structured Foreground Constraints

Sajid Javed[1], Seon Ho Oh[1], Andrews Sobral[2], Thierry Bouwmans[2], and Soon Ki Jung[1]
[1]Virtual Reality Lab, Kyungpook National University, Republic of Korea
[2]Laboratoire MIA (Mathematiques, Image et Applications)- Université de La Rochelle, France
[1]{sajid,shoh@vr.knu.ac.kr,skjung@knu.ac.kr},[2]{andrews.sobral,thierry.bouwmans@univ-lr.fr}

## Abstract

*Background subtraction process plays a very essential role for various computer vision tasks. The process becomes more critical when the input scene contains variation of pixels such as swaying trees, rippling of water, illumination variations, etc. Recent methods of matrix decomposition into low-rank (e.g., corresponds to the background) and sparse (e.g., constitutes the moving objects) components such as Robust Principal Component Analysis (RPCA), have been shown to be very efficient framework for background subtraction. However, when the size of the input data grows and due to the lack of sparsity-constraints, these methods cannot cope with the real-time challenges and always show a weak performance due to the erroneous foreground regions. In order to address the above mentioned issues, this paper presents a superpixel-based matrix decomposition method together with maximum norm (max-norm) regularizations and structured sparsity constraints. The low-rank component estimated from each homogeneous region is more perfect, reliable, and efficient, since each superpixel provides different characteristics with a reduced value of rank. Online max-norm based matrix decomposition is employed on each segmented superpixel to separate the low rank and initial outliers support. And then, the structured sparsity constraints such as the generalized fussed lasso (GFL) are adopted for exploiting structural information continuously as the foreground pixels are both spatially connected and sparse. We propose an online single unified optimization framework for detecting foreground and learning the background model simultaneously. Rigorous experimental evaluations on challenging datasets demonstrate the superior performance of the proposed scheme in terms of both accuracy and computational time.*

## 1. Introduction

Accurate and perfect segmentation of objects from videos is a complex task for many image processing and computer vision applications such as video surveillance, segmentation, compression, and scene understanding [11]. The most popular approach to accomplish this task is background subtraction (also called foreground detection). This approach requires the estimation of an accurate background model but it is challenging to design such a robust background subtraction method due to the presence of undesirable variations in pixel values, such as swaying trees, water surface, abruptly changing lighting conditions, etc. Consequently, the system always shows a weak performance in real-time scenarios.

Indeed, many methods have been devised to cope with the problems of background/foreground segmentation [1, 20]. Among them, the study of subspace learning models such as RPCA [3] have been attracted a lot of attention. RPCA decomposes the original data matrix $\mathbf{X}$ into *low-rank* $\mathbf{L}$ and *sparse* $\mathbf{S}$ components. The background sequence is then modeled by low-dimensional subspace having intrinsic structure called $\mathbf{L}$ and moving objects belong to the $\mathbf{S}$ component. But, RPCA currently suffers from some prominent issues [9]. First, these methods consider the entire video sequence as a vectorized data matrix for batch optimization processing. That is, the observation data must be stored in memory for the computation of *Singular Value Decomposition* (SVD) and hence arises the memory and computational challenges. Second, as the foreground pixels e.g $\mathbf{S}$ have small area in comparison with the background scene. Thus, without considering any structural contiguous constraints, the results of foreground detection always contain holes as well as the outliers noise.

To overcome the aforementioned limitations of RPCA, this paper presents a superpixel-based background subtraction algorithm with online matrix decomposition using max-norm constraints and efficient GFL to quest for intact structured foregrounds. We briefly summarize our methodology here. First, the superpixels are obtained from video frames and then, the static/dynamic homogeneous regions are classified. Second, iterative matrix decomposition is applied on each superpixel, to get more accurate estimation of **L** with reduced value of rank from each homogeneous region. We then use the max-norm constraints on the **L** component to prune majority of the outliers from estimated **L**. After that, the residual error is computed using the pre-computed **L**. To model the homogeneous perturbations of **L** and structural contiguities of **S**, the efficient GFL is finally exploited for continuous structural information. The GFL is a stable flexible structure prior for modeling the foreground objects (as it strengthens the fusion among the adjacent pixels) in a background subtraction problem. We propose an online formulation within a unified optimization framework to estimate **L** and detect the foreground objects at the same time.

The remainder of this paper is organized as follows. Section 2 summarizes the related work on RPCA-based matrix decomposition methods. Section 3 describes the proposed unified optimization framework for both online learning of **L** and **S**. Experimental analysis is presented in detail in section 4, and finally section 5 concludes our work.

## 2. Related Work

Background subtraction using matrix decomposition boasts of an extensive literature. Oliver *et al.* [16] presented one of the first proposal to model the background using *Principal Component Analysis* (PCA). The background sequence is then modeled by projecting the eigen-values when a new frame arrives. However, this model is not robust when an increasing number of outliers appear in a new subspace.

Cande's *et al.* [3] extended the PCA model by defining a more robust framework called RPCA via *Principal Component Pursuit* (PCP). Under some mild conditions, PCP perfectly recovers the **L** and **S** components. An excellent survey for background subtraction using RPCA-based matrix decomposition is summarized in [2].

However, the methods presented in [2] always show some noise since no additional contiguous constraints are considered on the residual error. There are several earlier works for video background subtraction based on unified matrix decomposition with additional constraints in **S**. For example, Zhou *et al.* [27] designed the *Detecting Contiguous Outliers in the Low Rank Representation* (DECOLOR) method to learn the background and then apply the *Markov Random Field* (MRF) model on the **S** matrix. But, due to the batch processing it is not desirable to process a large number of video frames.

In order to overcome this limitation, Feng *et al.* [5] proposed stochastic RPCA, and Javed *et al.* [9] adopted this method to model the **L** component and applied MRF to improve the foreground segments. However, this model is not performed within a single optimization framework. Thus, the reported performance is not competitive as compared to the other improved methods. For instance, Shakeri *et al.* [18] designed *Contiguous Outliers Representation via Online Low-rank Approximation* (COROLA) to improve the method presented in [9] using a unified optimization technique. The performance is encouraging only in case of dynamic background subtraction but the Gaussian Mixture Model (GMM) [20] is used to improve the **S** segments and then MRF constraints are applied.

Learning the **S** constraints has attracted a lot of attention as it improves the sparse structural priors [4, 22, 23, 24]. For example, Xin *et al.* [23] recently proposed a very interesting methodology using efficient GFL [22]. A superior performance is reported as compared to others [4, 9, 18, 24, 27]. However, an Augmented Lagrange Multiplier (ALM) batch optimization method is used for matrix decomposition. Moreover, two different algorithms are adopted. For example, *Singular Value Thresholding* (SVT) is applied for *Unsupervised Model Learning* (UML) when background/foreground coexist in each frame. A *Fast Iterative Soft Thresholding Algorithm* (FISTA) is applied for *Supervised Model Learning* (SML) case when pure background frames are available. Furthermore, the method is computationally expensive since it is based on batch strategy which hardly process more than 400 frames with image resolution of $[240 \times 320 \times 3]$.

In contrast to the aforementioned techniques above, our proposed superpixels based background/foreground segmentation is more efficient, effective, and online as compared to previous methods [4, 23]. We propose an online formulation to learn the **L** and structural **S** components in a single optimization scheme. We use only one algorithm called max-norm based online matrix decomposition scheme that processes each homogeneous region from one frame per time instance to separate the **L** and **S** components for UML and SML case and then the sparsity structure is learnt using an adaptive version of efficient GFL [22] with a fast parametric flow method [6].

## 3. Proposed Methodology

In this section, we present our proposed algorithm in detail for background subtraction. The proposed scheme consists of several stages which are described in the following sections.

### 3.1. Superpixel Segmentation

Contrary to [23, 25], where the entire video sequence is decomposed into **L** and **S** components and thus the computational issues arise. We start our designed scheme with superpixel segmentation to separate **L** and **S** from each homogeneous region. Each **L** and **S** components are likely to be homogeneous hence the estimated **L** (i.e., the background model) is to be more reliable and accurate given a limited number of pixels. In this paper, the *Entropy Rate Superpixel Segmentation* (ERS) [13] method is considered due to its efficiency, simplicity, and good performance. In ERS, the superpixel segmentation is considered as a graph partitioning problem. Given a graph **G**=(**V**, **E**) and the number of superpixels $k$, the goal is to find out the subset of edges, e.g., **A**⊆**E** such that the resulting graph $\hat{\mathbf{G}}$=(**V**, **A**) contains $k$ connected subgraphs. **V** is the vertex (e.g., corresponds to the pixels in image) and **E** is the edges typically constructed by the 4-neighborhood system. In addition, when an edge is not included in **A** then its weight is computed by the similarity between the features observed at the connected vertices. The objective function to solve the graph partitioning problem is then given by

$$\max_{\mathbf{A}} \mathcal{H}(\mathbf{A}) + \lambda\mathcal{B}(\mathbf{A}) \\ such\ that\ \mathbf{A} \subseteq \mathbf{E}\ and\ N_A \geq k, \tag{1}$$

where $\mathcal{H}(\cdot)$ and $\mathcal{B}(\cdot)$ denote the entropy rate of random walk and balancing term, respectively, and $N_A$ is the number of connected components in **G**. $\mathcal{H}(\cdot)$ makes compact and homogeneous regions more stable, whereas the $\mathcal{B}(\cdot)$ segments the superpixels with similar sizes. The exact optimization of the graph is difficult to maximize, however, this problem can be solved via an efficient greedy algorithm designed in [13] that provides almost 0.5 approximation bound. Interested readers may further explore the details about ERS in [13].

Within the context of the background scene, the superpixels may consist of static (i.e., no variations in pixel value) and dynamic (i.e., swaying trees and water rippling) regions. In this paper, we separately deal with the static and dynamic superpixels for online matrix decomposition to reduce the computational load. As the value of rank is different for these regions therefore we first identify these regions by considering a fixed window size of *N*, in which $\mathbf{X}^t$ and $\mathbf{X}^{t+1}$ be the sets of superpixels computed, respectively, at frame *t* and $t+1$. Then, the difference is computed between each consecutive superpixels, e.g., $\mathbf{x}_k^t \in \mathbf{X}^t$ and $\mathbf{x}_k^{t+1} \in \mathbf{X}^{t+1}$, where $\mathbf{x}_k$ is the $k^{th}$ superpixel of **X**. If the difference is above a threshold $\epsilon$, we mark this homogeneous region as dynamic superpixel. The threshold $\epsilon$ is adaptively computed as the average of the difference between all the superpixels in frame $t+1$. Fig. 1 (c)-(d) show the segmentation of superpixels of *Water Surface* sequence taken from the *Perception Test Images Sequences* (PTIS) dataset [10].

### 3.2. Background/Foreground Model

In this section, we present our online background with structural foreground modeling scheme in detail.

Let say that $\mathbf{X}^t$ be a set of input superpixels (computed in the previous section) at a time *t*, which is corrupted by *outliers* say **S**, then **X** can be reconstructed by the summation of background model **L** and foreground **S**, e.g., **X**=**L**+**S**. If we consider the **L** structure of background and structured sparisty of **S**, then this optimization problem can be written as

$$\min_{\mathbf{L},\mathbf{S}} \frac{1}{2}||\mathbf{L}||_{max}+\beta \underbrace{||\mathbf{S}||_1+\gamma||\Phi(\mathbf{S})||_1}_{GFL} \\ such\ that\ \mathbf{X} = \mathbf{L} + \mathbf{S}. \tag{2}$$

When **X**, **L**, or **S** is considered, it means that each superpixel is being processed in **X**, **L**, or **S** at a time *t*. As compared to its batch counterpart [23], we use the max-norm constraints such as $||\mathbf{L}||_{\max}$ in Eq. 2 to promote the *low rank* structure in each superpixel, as it is more superior than the nuclear norm as described in [8, 19] when most of the entries are corrupted in observation data. $\beta$ and $\gamma$ are constant paramters which are estimated online during **S** component optimization. $||\mathbf{S}||_1$ is the observed data in Eq. 2 which imposes the sparsity constraints on the foreground, such that the foreground pixels should be small. The third term $||\Phi(\mathbf{S})||_1$ in Eqn. 2 represents the difference between the adjacent pixels, which is computed as

$$||\Phi(\mathbf{S})||_1 = \sum_{(i,j)\in\mathcal{N}} \mathbf{w}_{ij}^t |\mathbf{s}_i^t - \mathbf{s}_j^t|, \tag{3}$$

where $\mathcal{N}$ is a neighborhood system on pixels. $||\Phi(\mathbf{S})||_1$ measures the cost of assigning the labels $s_i$ and $s_j$ to the neighboring pixles $i$ and $j$, respectively. The $\mathbf{w}_{ij}$ in Eq. 3 is the adaptive weighting factor between the pixels, and it makes the fusion more stable between the neighboring pixels as

$$\mathbf{w}_{ij}^t = \exp \frac{||\mathbf{y}_i^t - \mathbf{y}_j^t||_2^2}{2\sigma^2}, \tag{4}$$

where **y** is the pixel intensity and $\sigma$ is a tunning parameter which will be discussed later. In [23], $\mathbf{w}_{ij}^t$ is computed for only test sequence, whereas in this study it is computed for each homogeneous region of every frame to adapt the changes in a background intensity. Basically, the GFL [22] finds the continuous and small varaitions of outliers to represent the foreground mask due to the $l_1$-norm or penalty on each adjacent pixels. Eq. 2 is the main equation of our model which is non-convex and needs to be solved via online manners for real-time systems. Earlier approaches such as [23, 24, 27] solved the problem under batch processing umbrella, where the entire data is processed for the computation of SVD. Here, we solve this equation using online estimation of **L** from each superpixel in **X**, and then impose the structural constraints. Each superpixel of one frame is processed per time instance. We solve the above equation in the following steps.

#### 3.2.1 Estimation of Low-rank Component

Although Eq. 2 completely fits to model the proposed method, but optimization is the major challenge specially when it contains two different norms and the data size grows. We use the iterative matrix decomposition method [19] using the max-norm presented in [8] to separate the $\mathbf{L}$ given the initial approximation of $\mathbf{S}$, Eq. 2 can be re-written as

$$\min_{\mathbf{L}} \frac{1}{2}||\mathbf{X}-\mathbf{L}-\mathbf{S}||_F^2+\lambda_1||\mathbf{L}||_{\max}^2. \tag{5}$$

Since all the samples are tightly coupled in $||\mathbf{L}||_{max}$, therefore these samples are accessed during optimization at each iteration which prevents them from processing big data. In constrast, an equivalent form of max-norm designed in [19] is used in this work, whose rank is upper bounded by $d$ as

$$\begin{aligned} ||\mathbf{L}||_{\max}= \min_{\mathbf{U}\in\mathbb{R}^{p\times d},\mathbf{V}\in\mathbb{R}^{n\times d}} \frac{1}{2}(||\mathbf{U}||_{2,\infty}\cdot||\mathbf{V}||_{2,\infty}) \\ \textit{such that } \mathbf{L}=\mathbf{U}\mathbf{V}^T, \end{aligned} \tag{6}$$

where $p$ denotes the dimension of each superpixel for each sample. For instance, $\mathbf{x}_k$ is the $k^{th}$ superpixel in a set $\mathbf{X}$ and its dimension is $p$, i.e., $\mathbf{x}_k \in \mathbb{R}^p$, $n$ is the number of samples and $d$ is a rank. $||\mathbf{U}_i||_{2,\infty}$ and $||\mathbf{V}_i||_{2,\infty}$ are the maximum $l_2$ row norms of basis and coefficients. But, the coefficients must be positive and therefore $||\mathbf{V}||_{2,\infty}^2= 1$ as proved in [19].

Eq. 6 shows that each homogeneous region in $\mathbf{L}$ matrix can be an explicit product of each low-dimensional subspace basis $\mathbf{U} \in \mathbb{R}^{p\times d}$ and its coefficient $\mathbf{V} \in \mathbb{R}^{n\times d}$ and this re-formulated max-norm is shown in recent works [8, 19]. In other words, it seems that the background model $\mathbf{L}$ is represented by the linear combination of a small number of the columns of $\mathbf{U}$. Hence, Eq. 5 is re-formulated by substituting Eq. 6 for objective function minimization as

$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}||\mathbf{X}-\mathbf{U}\mathbf{V}^T-\mathbf{S}||_F^2+\frac{\lambda_1}{2}||\mathbf{U}||_{2,\infty}^2, \tag{7}$$

where $\lambda_1$ is a regularization parameter for $\mathbf{L}$ component patterns. Eq. 7 is the main equation for online matrix decomposition of all video frames, which is not completely convex with respect to $\mathbf{U}$ and $\mathbf{V}$. More specifically, since we process each superpixel (static or dynamic) of one video sample per time instance $t$, then Eq. 7 can be reduced into more compact form in case for each superpixel per sample as

$$\min_{\mathbf{U},\mathbf{v}} \frac{1}{2}\sum_{t=1}^{n}||\mathbf{x}_k^t-\mathbf{U}^t\mathbf{v}_k^t-\mathbf{s}_k^t||_2^2+\frac{\lambda_1}{2}||\mathbf{U}||_{2,\infty}^2. \tag{8}$$

In this case, $\mathbf{x}_k^t$ is the $k^{th}$ superpixel at a time $t$, $\mathbf{U}^t$ is its basis matrix, and $\mathbf{v}_k^t$ is a coefficients vector. Alg. 1 summarizes the proposed online decomposition by processing one frame per time instance. We briefly explain the main notion of each step of Alg. 1. The coefficient $\mathbf{v}$ and basis $\mathbf{U}$ for each superpixel are optimized in an iterative way.

### Solving Coefficients v

First, the coefficients vector $\mathbf{v}$ is estimated with fixed random basis $\mathbf{U}$ (in case of SML/UML only a small number of frames are required) by projecting one frame at a time $t$. In Alg. 1, the step 3 requires the convex optimization problem for computing $\mathbf{v}$ as

$$\mathbf{v}^t = (\mathbf{U}^T\mathbf{U}+\alpha\mathbf{I})^{-1}\mathbf{U}^T\left\{\mathbf{x}_k^j-\mathbf{s}^{j-1}\right\}, \tag{9}$$

where $\alpha$ is the positive dual variable. In the next iteration, if $||\mathbf{v}||_2\leq 1$, then $\mathbf{v}$ will remain same, otherwise it will be updated as

$$\mathbf{v} = \underset{\alpha,\alpha>0,\mathbf{v}}{\operatorname{argmax}} \min_{||\mathbf{v}||_2=1} \frac{1}{2}||\mathbf{x}_k-\mathbf{U}\mathbf{v}_k-\mathbf{s}_k||_2^2+\frac{\alpha}{2}(||\mathbf{r}||_2^2-1). \tag{10}$$

Basically, Eq. 9 is the closed form solution of $\mathbf{v}$ of Eq. 10. More details can be found in [8, 19].

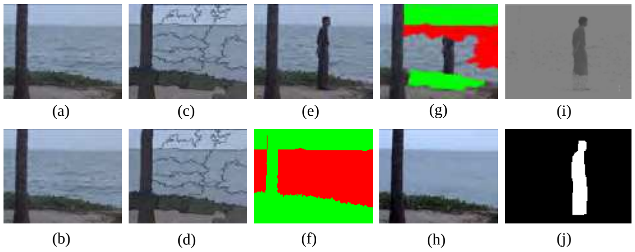
### Solving Basis U

The basis $\mathbf{U}^t$ for each superpixel per frame is estimated at step 5 of Alg. 1 through minimizing the previously computed coefficients $\mathbf{v}$. These basis $\mathbf{U}^t$ for low-dimensional subspace learning is then updated through block coordinate decent method by the result of previously computed $\mathbf{U}$. The basis vector $\mathbf{U}$ is updated column-wise as

$$\mathbf{u}_j \leftarrow \mathbf{u}_j - \frac{1}{\mathbf{P}_{jj}}(\mathbf{U}\mathbf{p}_j-\mathbf{q}_j+\lambda_1\mathbf{l}_j), \tag{11}$$

where $\mathbf{u}_j$ is the $j^{th}$ column of the basis vector $\mathbf{U}$ for each homogeneous region, and $\mathbf{p}$ and $\mathbf{q}$ correspond to the accumulation matrices $\mathbf{P}$ and $\mathbf{Q}$, respectively. The $\mathbf{l}_j$ is the $j^{th}$ column of the subgradient of basis $\mathbf{U}$. If the rank $d$ is given and basis $\mathbf{U}$ is estimated as above which is a fully $d$, then $\mathbf{U}$ converges to the optimal solution asymptotically as compared to its batch counterpart shown in [5]. Since the $\mathbf{U}$ is updated colum-wise therefore it is independent to the number of samples and hence it solves the computational issues. Finally, the background sequence is then learnt by $\mathbf{L}$ matrix that is the multiple of basis and its coefficients which changes sequentially at a time instance $t$ as presented in Fig. 1 (h).

First, we analyse the issues by applying the same method on a global frame, then we describe the characteristics of applying it on a superpixel regions. Fig. 2 shows the results after applying the scheme on a global frame. The two types of background scenes such as static called *office* from CD-net [21] and dynamic called *WaterSurface* from PTIS [10] are considered for $\mathbf{L}$ component estimation with different values of $d$. With a higher value of rank such as $d = 5$, the outliers appear on a static region which degrades the accuracy. In addition, the computational issues arise due to the



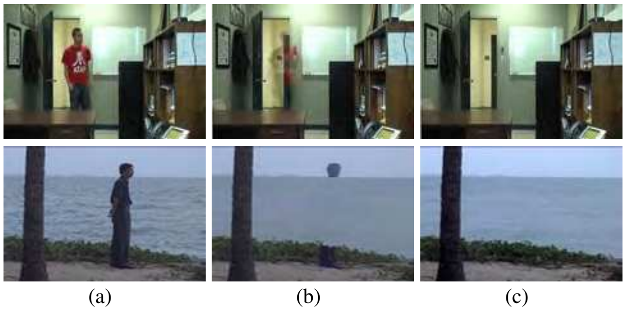
(a) (c) (e) (g) (i)

(b) (d) (f) (h) (j)

Figure 1. Schematic example of our background subtraction scheme and overview of superpixels segmentation with $rank$ selection. (a) and (b) two input frames, (c) and (d) the corresponding superpixels segmentation of (a) and (b). (e) test frame, (f) identification of static/dynamic homogeneous regions (green denotes the static whereas the red one represents the dynamic superpixels), (g) few superpixels on test image, (h) the exact recovery of $\mathbf{L}$ matrix, (i) the residual error $\hat{\mathbf{S}}$, and (j) the result of GFL.

size of basis $\mathbf{U}$ increased. In contrast, the $d$ is quite enough for dynamic regions for subspace update to get more accurate $\mathbf{L}$ component. Fig. 2 (b) demonstrates this problem more clearly. On the other hand, taking a smaller value of $d = 1$, the static scene provides an exact estimation of $\mathbf{L}$ matrix, however, it is not quite enough for the dynamic regions to update most of the scene, hence the noise occurs in a foreground mask.

Using static/dynamic superpixel classification, each region is processed using different values of $d$, since the static and dynamic regions may occur simultaneously. Fig. 1 shows the accurate recovery of $\mathbf{L}$ by considering $d = 1$ for static superpixel, and $d = 5$ for dynamic homogeneous region. Using these adapted values of $d$ on separate regions improve the processing time as the number of columns in $\mathbf{U}$ reduces. However, when the foreground is distributed among on or more homogeneous regions, the system shows a weak performance since the holes appear due to the lack of compactness but it can be solved by exploiting the structural foreground constraints.



(a) (b) (c)

Figure 2. An example of different types of background scenes. (a) input, (b) $\mathbf{L}$ component using $d = 5$, and (c) $\mathbf{L}$ component using $d = 1$. From top to bottom, *Office* (static) and *WaterSurface* (dynamic) sequence.

### 3.2.2 Outliers Estimation

In this section, we describe the structural foreground constraints imposed on the $\mathbf{S}$ component. Since we computed the background model $\mathbf{L}$ in the previous section by changing Eq. 2 into Eq. 5 with max-norm regularizations. For continuous variations in $\mathbf{S}$, we then re-write the Eq. 2 by introducing the initial residual error $\hat{\mathbf{S}}$ as

$$\min_{\mathbf{S}} \frac{1}{2}||\hat{\mathbf{S}}||_F^2 + \beta||\mathbf{S}||_1 + \gamma||\Phi(\mathbf{S})||_1 = \\ \sum_i \hat{\mathbf{S}}_i^2 + \beta\sum_i \mathbf{s}_i + \gamma\sum_{(i,j)\in\mathcal{N}} \mathbf{w}_{ij}|\mathbf{s}_i - \mathbf{s}_j|, \qquad (12)$$

and in case of each superpixel as

$$\min_{\mathbf{s}_k} \frac{1}{2}||\hat{\mathbf{s}}_k||_F^2 + \beta||\mathbf{s}_k||_1 + \gamma||\Phi(\mathbf{s})||_1, \qquad (13)$$

where $\hat{\mathbf{S}}$ is the residual error computed by $\hat{\mathbf{S}} = \mathbf{X} - \mathbf{L}$, as depicted in Fig. 1 (i), and $\mathbf{L}$ is the set of *low-rank* superpixels estimated in the previous section. The first term in Eq. 12 is constant, second and third terms correspond to the GFL for continuous structural contiguity of the foreground objects (since the foreground pixels are too small and connected).

The goal is to find the $\mathbf{S}$, and this problem can be solved through minimizing the energy in Eq. 12 by considering that the graph having nodes $\mathbf{V}$ and edges $\mathbf{E}$, where each variable corresponds to a node on the graph.

The $\sigma$ in Eq. 4 is a smoothing term, which must be tunned carefully, otherwise the model in Eq. 2 reduces to the traditional RPCA [3]. A fixed value in [23] is considered for each test sequence and thus cannot be adapted for every scene. In this proposal, we update it according to the foreground object. The $\sigma$ is updated as $\sigma + \frac{|(X-UV)^2|_1}{2}$. The $\frac{|(X-UV)^2|}{2}$ will be higher in the presence of foreground object but without any object in the scene its value will be lower.

Eq. 12 is similar to the parametric graph-cut problem and its optimization is difficult to solve. We use the more greedy method proposed in [22] to solve this equation. This method is fast enough as the element-wise soft thresholding is applied and then, update the foreground with a fast

**Algorithm 1** Iterative Max-Norm Decomposition

```
Input: $\mathbf{X}$ (set of input superpixels), $\mathbf{S} = 0$, $\mathbf{U} \in \mathbb{R}^{p \times d}$(initial basis), $d$, $\mathbf{P} \in \mathbb{R}^{r \times r}$, $\mathbf{Q} \in \mathbb{R}^{p \times r}$, $\mathbf{v} \in \mathbb{R}^{d}$, $\mathbf{V} \in \mathbb{R}^{n \times d}$,Unitary Matrix $\mathbf{I}$, $\lambda_1$.
for $t = 1$ to $n$ do {Access each sample}
    for $i = 1$ to $k$ do {each superpixel}
        Compute the initial coefficients v by projecting the new sample as
        $\mathbf{v}^t = arg \min_{\mathbf{v}, ||\mathbf{v}||_2^2 \leq 1} \frac{1}{2} ||\mathbf{x}_k^t - \mathbf{U}^{t-1}\mathbf{v} - \mathbf{s}^t||_2^2$
        V(t,:) $\leftarrow \mathbf{v}^t$. Compute the auxiliary matrices $\mathbf{P}^t$ and $\mathbf{Q}^t$ as
        $\mathbf{P}^t \leftarrow \mathbf{P}^{t-1} + \mathbf{v}\mathbf{v}^T, \mathbf{Q}^t \leftarrow \mathbf{Q}^{t-1} + (x_k^t - \mathbf{s}^t)\mathbf{v}^T$
        Compute $\mathbf{U}^t$ with $\mathbf{U}^{t-1}$ as
        $\mathbf{U}^t = argmin \frac{1}{2} Tr[\mathbf{U}^T(\mathbf{P}^t + \lambda_1 \mathbf{I})\mathbf{U}]$
        $-Tr(\mathbf{U}^T\mathbf{Q}^t) + \frac{\lambda_1}{2}||U||_{2,\infty}^2$
        Update the basis U by Block Coordinate Descent method.
        $\mathbf{L}^t \leftarrow \mathbf{U}\mathbf{V}^T$ (set of superpixels of low-rank)
        $\hat{\mathbf{S}} = \mathbf{X} - \mathbf{L}$, initial error
        Optimize S using pre-defined $\hat{\mathbf{S}}$ as
        $\mathbf{S} \leftarrow arg \min_{\mathbf{s}} \beta \sum_i \mathbf{s}_i + \gamma ||\Phi(\mathbf{S})||_1$
    end for
end for
Output: L, S.
```

parametric-flow algorithm proposed in [6] to reinforce most of the foreground pixels and smoothness. Applying GFL on each small homogeneous region, we achieved more good computational time as compared to [23]. The result of GFL is a foregorund mask as shown in Fig. 1 (j).

# 4. Experimental Evaluations

In this section, the detailed experiments including both visual and numerical results are presented. First, we discuss the implementation setup and then, the datasets with results and computational complexity of the algorithm is described with discussion.

## 4.1. Implementation Details

Our algorithm is implemented on Matlab R2013a with 3.40 GHz Intel core i5 processor with 4 GB RAM. Since, the proposed method is based on superpixel processing, therefore the regularization parameter such as $\lambda_1$ in Eq. 7 is considered for each homogeneous region as $\lambda_1 = \frac{0.25}{\sqrt{\max(size(\mathbf{x}_k))}}$. We used the number of superpixels in Eq. 1 as $k = 20$ and window size of $N = 100$ for static/dynamic superpixel identification. The positive value should be used for smoothing factor $\sigma$ in Eq. 4 for $\mathbf{S}$ component optimization. We initially set its value as $\sigma = 25$ and then, it is updated as mentioned earlier. Moreover, we have not used any post-processing such median filtering to remove small moise from binary mask.

## 4.2. Datasets

We have tested our designed methodology on one small dataset called *Perception Test Images Sequences* (PTIS) [10] and second a very large scale dataset called *Change Detection* (CDnet 2014) [21]. The brief description about these datasets are summarized below.

### 4.2.1 Perception Test Images Sequences Dataset

PTIS dataset [10] contains nine complex challenging background scenes. The $1^{st}$ five SML sequences called, *Campus* (CAM), *Fountain* (FT), *Water Surface* (WS), *Moving Curtain* (MC), and *Lobby* (LB), respectively, belongs to the highly dynamic backgrounds. The remaining four sequences called, *Shopping Mall* (SM), *Airport Hall* (AH), *Restaurant* (RT), and *Escalator* (ES) corresponds to the bootstrapping case e.g., UML. The size of each sequence is $[128 \times 160 \times 3]$ . Fig. 3 shows the visual results of PTIS dataset.

Due to space limitations, we have not provided the detailed qualitative comparison with other approaches. However, we have only presented the comparison of one sequence called LB with the most recent method called *Background Subtraction via Generalized Fussed Lasso Foreground Modeling* (BS-GFLFM) [23]. LB is a dynamic scene, which contains slowly as well as sudden changing lighting conditions. It also contains some training background frames for model learning e.g., SML. Since, two different algorithms are used for SML and UML cases in [23] and most of the time the pure background frames are less representative for lighting conditions. In contrast, we have used only one algorithm for SML/UML cases and the proposed updating method for basis **U** presented above handles the different illumination conditions. The author's of [23] reported only one sequence of LB for visual evaluations. The detailed comparison of LB scene can be depicted in Fig. 4.

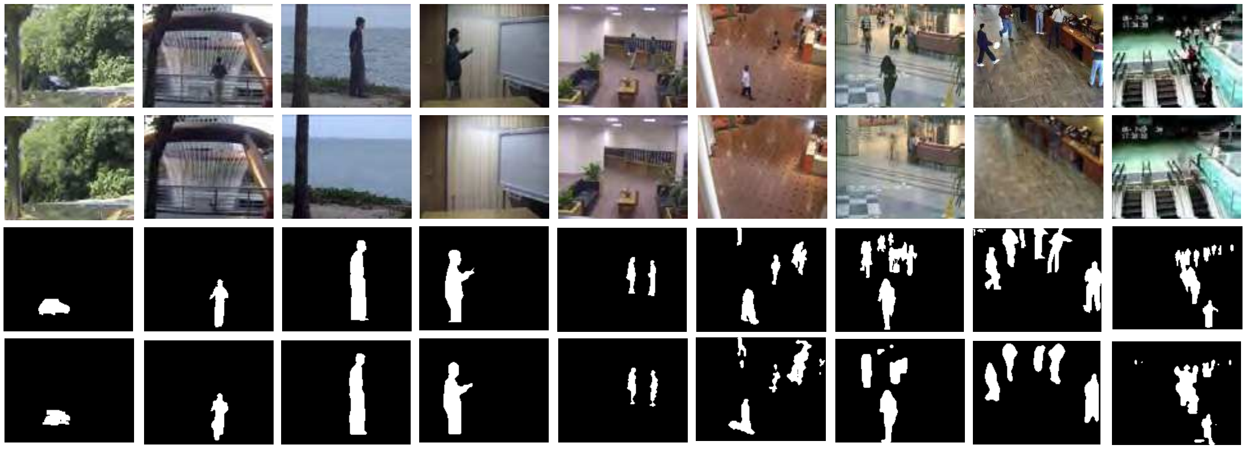
For numerical analysis, the proposed method is compared with eight state-of-the-art approaches which can be categorized into two categories. Since we have employed the sparsity-constraints, therefore OR-PCA with MRF [9], COROLA [18], DECOLOR [27], BS-GFLFM [23], GOSUS[1] [24], and LR-FSO[2] [25] methods are considered first for comparison. Second, traditional RPCA [3], and SemiSoftGoDec [26] schemes are also taken into consideration. We use the well-known measure called *F-measure* for evaluation. It is computed by comapring each sequence with its available ground truth frame. Table. 1 shows that

[1] Grassmannian Online Subspace Updates with Structured-sparisty.

[2] Linear Regression model with Fused Sparsity on Outliers.

Table 1. Qualitative results of PTIS dataset [10]: Average *F-measure* score of each video sequence with earlier approaches.

| **Method** | **Perception Test Images Sequences dataset** [10] | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | CAM | FT | WS | MC | LB | SM | AH | RT | ES | Average |
| Evaluation Frames | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | |
| OR-PCA with MRF [9] | 0.7597 | 0.8223 | 0.9166 | 0.8920 | 0.8081 | 0.8072 | 0.7844 | 0.7150 | 0.6468 | 0.7946 |
| COROLA [18] | 0.7706 | 0.9175 | 0.9503 | 0.9038 | 0.8125 | 0.8298 | 0.7955 | 0.7320 | 0.6603 | 0.8191 |
| DECOLOR [27] | 0.3416 | 0.2075 | 0.9022 | 0.8700 | 0.6460 | 0.6822 | **0.8169** | 0.6589 | 0.7480 | 0.6525 |
| LR-FSO [25] | 0.7457 | 0.7760 | 0.8554 | 0.8136 | 0.6911 | 0.6844 | 0.6296 | 0.5798 | 0.6102 | 0.7095 |
| GOSUS [24] | 0.8347 | 0.87890 | 0.9236 | 0.8995 | 0.6996 | 0.8019 | 0.5616 | 0.7475 | 0.6432 | 0.7767 |
| BS-GFLFM [23] | 0.8126 | 0.9011 | 0.9366 | 0.9455 | 0.6456 | **0.8326** | 0.7422 | 0.8265 | 0.7613 | 0.8226 |
| RPCA [3] | 0.5226 | 0.8650 | 0.6082 | 0.9014 | 0.7245 | 0.7785 | 0.5879 | 0.8322 | 0.7374 | 0.7286 |
| SemiSoftGoDec [26] | 0.0903 | 0.2574 | 0.4473 | 0.4344 | 0.3602 | 0.6554 | 0.5713 | 0.3561 | 0.2751 | 0.3830 |
| Ours | **0.8574** | **0.9322** | **0.9856** | **0.9744** | **0.8840** | 0.8265 | 0.7739 | **0.8394** | **0.8029** | **0.8751** |



(a) CAM (b) FT (c) WS (d) MC (e) LB (f) SM (g) AH (h) RT (i) ES

Figure 3. Visual results on PTIS dataset. From top to bottom, input, *low-rank*, ground truth, and the results of proposed method.

the proposed method out-performs with earlier methods. It should be noted that the author's of BS-GFLFM [23] reported thier performance using one test sequence. In this proposal, we have reported the performance on all the sequences and then, average score is computed as shown in Table. 1.

### 4.2.2 Change Detection 2014 Dataset

CDnet [21] is the large scale real-world dataset. It contains about 55 video sequences divided into 11 different categories. The proposed algorithm is evaluated on 5 simple as well as complex categories namely: *Baseline*, *Dynamic Background*, *Intermittent Object Motion*, *Thermal*, and *Low Framerate*. The size of each video frame is $[240 \times 320 \times 3]$ and the number of frames vary from $1,000$ to $6,000$.

We showed the visual results on selected sequences such as *highway*, and *office* taken from *Baseline* category. The *canoe*, and *overpass* sequences are taken from *Dynamic Background*, *sofa* sequence is selected from *Intermittent Object Motion*, *library* from category *Thermal*, whereas the *turnpike* sequence is chosen from *Low Framerate* category. Fig. 5 presents the visual performance of these sequences.

The proposed scheme is also evaluated for quantitative analysis on CDnet. We compute the same *F-measure* score as discussed previously. For example, the *office* sequence of *Baseline* categroy contains $2,050$ frames and the evaluation is required from 570 to $2,050$ number of frames. Since most of the RPCA-based and structured-sparsity methods process under batch strategy, and thus these methods can not process such a large dimenisonal data. Therefore, we compare our quantitative performance with the most recent real-time methods such as SOBS-CF [15], CwisarDH [7], CP3-Online [12], Multiscale Model [14], and Spectral-360 [17] as reported on CDnet benchmark [21]. Table. 2 shows the achieved performance with other methods. The (**-**) lines in Table. 2 demonstrates that the method can not process a huge data such as $[240 \times 320 \times 500]$. Moreover, according to the results reported on CDnet [21] site, we are the $4^{th}$ top performer in *Baseline*, and *Thermal*. The $3^{rd}$ performer in *Dynamic Background*, and *Low Framerate*, and finally $6^{th}$ performer in case of *Intermittent Object Motion* category. The detail quantitative performance can be found on the benchmark site.

## 4.3. Computational Time

Time is also evaluated during the experiments and it is reported in CPU time. For fair comparison with other methods [23, 24], we first make a batch of 100 frames with a resolution of $120 \times 160$, e.g., $[120 \times 160\ \ 100]$ and then,

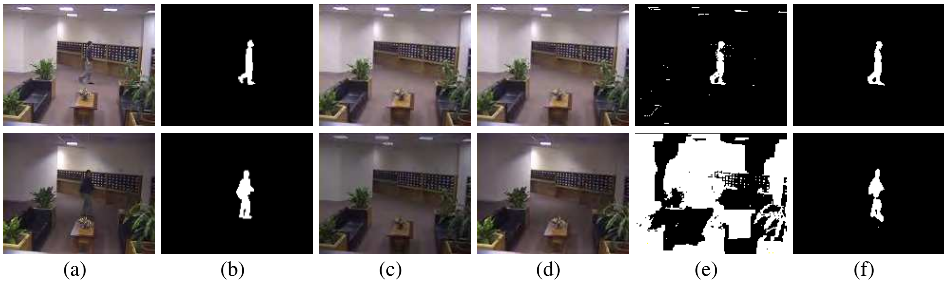


Figure 4. LB sequence of PTIS dataset. (a) two input scenes with different lighting conditions, (b) ground truth, (c) **L** matrix of [23], (d) **L** matrix of proposed method, (e) foreground mask of [23], and (f) mask of ours.

Table 2. Qualitative results of CDnet dataset [10]: Average *F-measure* score of each category with previous methods.

| **Method** | **Change Detection 2014 dataset** | | | | | |
|---|---|---|---|---|---|---|
| | Baseline | Dynamic Background | Intermittent Object Motion | Thermal | Low Framerate | Average |
| Number of Videos | 4 | 6 | 6 | 5 | 4 | |
| SOBS-CF [15] | 0.9299 | 0.6519 | 0.5810 | 0.7140 | 0.5148 | 0.6783 |
| CwisarDH [7] | 0.9145 | 0.8274 | 0.5753 | 0.7866 | 0.6406 | 0.7488 |
| CP3-Online [12] | 0.8856 | 0.6111 | 0.6177 | 0.7917 | 0.4742 | 0.67606 |
| Multiscale Model [14] | 0.8450 | 0.5953 | 0.4497 | 0.5103 | 0.3365 | 0.5473 |
| Spectral-360 [17] | 0.9330 | 0.7766 | 0.5609 | 0.7764 | 0.6437 | 0.7381 |
| BS-GFLFM [23] | - | - | - | - | - | - |
| GOSUS [24] | - | - | - | - | - | - |
| Ours | **0.9469** | **0.8519** | **0.6988** | **0.8156** | **0.6836** | **0.7993** |

$[240 \times 320 \;\; 100]$. We have tested the method using the same configurations as discussed above. Since each superpixel of a one frame is processed per time instance via online manners therefore it is indepedent of the number of images and thus time grows linearly as the image resolution grows. In both cases, Table. 3 demonstrates that the proposed method shows attractive computational time as compared to earlier methodologies. In case of large video data, the earlier approaches such as [23, 24] fail to process as mentioned in Table. 3. Although, the method is currently written in Matlab but we believe that these all experimental evaluations on the proposed scheme shows a very good speed/accuracy trade-off.

## 5. Conclusion

In this paper, we presented an online coarse-to-fine strategy for video background/foregorund segmentation. The proposed scheme is based on the processing of one frame per time instance. Our method provide an efficient, and more reliable *low-rank* component using matrix decomposition with max-norm of superpixels. Moreover, the foreground labels are maintained using the sparsity-constraints. Experimental results are the evidence that we have achieved a superior performance as compare to its batch counterparts. Our further investigations will concern the extension to the moving cameras case.

Table 3. Computational Time in seconds

| **Resolution × No. of Images** | GOSUS [24] | BS-GFLFM [23] | | Ours |
|---|---|---|---|---|
| | | SML | UML | |
| $[128 \times 160] \times 100$ | 69.77s | 88.07s | 340.12s | **8.12**s |
| $[240 \times 320] \times 100$ | - | - | - | **19.22**s |

## 6. Acknowledgments

This work is supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the C-ITRC (Convergence Information Technology Research Center) (IITP-2015-H8601-15-1002) supervised by the IITP (Institute for Information & communications Technology Promotion).

## References

[1] T. Bouwmans. Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, 11:31–66, 2014.

[2] T. Bouwmans and E. H. Zahzah. Robust PCA via Principal Component Pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, pages 22–34, 2014.

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust Principal Component Analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[4] X. Cui, J. Huang, S. Zhang, and D. N. Metaxas. Background subtraction using low rank and group sparsity constraints. In *Computer Vision–ECCV 2012*, pages 612–625. Springer, 2012.

[5] J. Feng, H. Xu, and S. Yan. Online Robust PCA via Stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 404–412, 2013.

[6] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55, 1989.

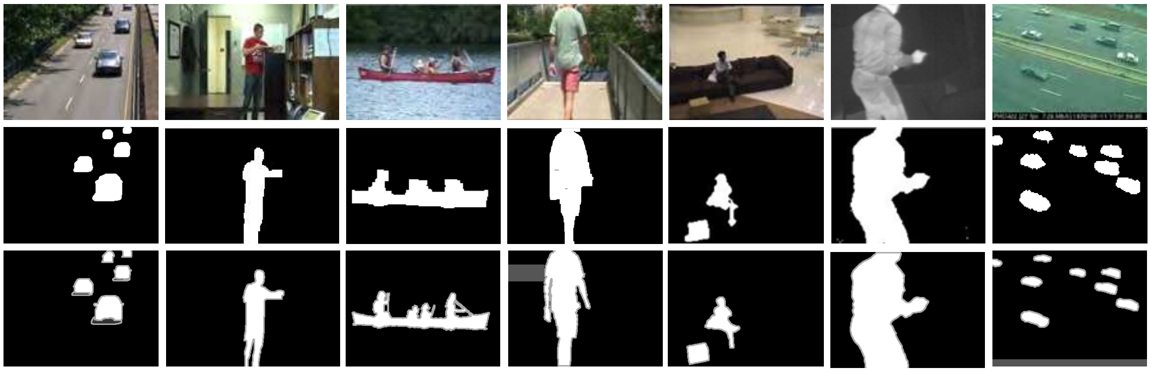


Figure 5. Qualitative results of CDnet dataset [21]. From top to bottom, input, ground truth, and the results of proposed method.

[7] M. D. Gregorio and M. Giordano. Change detection with weightless neural networks. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 409–413, 2014.

[8] A. Jalali and N. Srebro. Clustering using max-norm constrained optimization. *arXiv preprint arXiv:1202.5598*, 2012.

[9] S. Javed, S. H. Oh, A. Sobral, T. Bouwmans, and S. K. Jung. OR-PCA with MRF for Robust Foreground Detection in Highly Dynamic Backgrounds. In *Computer Vision–ACCV 2014*, pages 284–299. Springer, 2015.

[10] L. Li, W. Huang, I.-H. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *Image Processing, IEEE Transactions on*, 13(11):1459–1472, 2004.

[11] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2036–2043. IEEE, 2009.

[12] D. Liang and S. Kaneko. Improvements and experiments of a compact statistical background model. *arXiv preprint arXiv:1405.6275*, 2014.

[13] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2097–2104. IEEE, 2011.

[14] X. Lu. A multiscale spatio-temporal background model for motion detection. In *Image Processing, 2014 IEEE Conference on*, 2014.

[15] L. Maddalena and A. Petrosino. A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection. *Neural Computing and Applications*, 19(2):179–186, 2010.

[16] N. M. Oliver, B. Rosario, and A. P. Pentland. A Bayesian Computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831–843, 2000.

[17] M. Sedky, M. Moniri, and C. C. Chibelushi. Spectral-360: A physics-based technique for change detection. pages Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on, 2014.

[18] M. Shakeri and H. Zhang. COROLA: A Sequential Solution to Moving Object Detection Using Low-rank Approximation. *arXiv preprint arXiv:1505.03566*, 2015.

[19] J. Shen, H. Xu, and P. Li. Online optimization for Max-norm regularization. In *Advances in Neural Information Processing Systems*, pages 1718–1726, 2014.

[20] 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.*, volume 2. IEEE, 1999.

[21] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar. Cdnet 2014: An expanded change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 393–400. IEEE, 2014.

[22] B. Xin, Y. Kawahara, Y. Wang, and W. Gao. Efficient Generalized Fused Lasso and Its Application to the Diagnosis of Alzheimers Disease. 2014.

[23] B. Xin, Y. Tian, Y. Wang, and W. Gao. Background Subtraction via Generalized Fused Lasso Foreground Modeling. *CoRR*, abs/1504.03707, 2015.

[24] J. Xu, V. K. Ithapu, L. Mukherjee, J. M. Rehg, and V. Singh. Gosus: Grassmannian online subspace updates with structured-sparsity. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3376–3383. IEEE, 2013.

[25] G. Xue, L. Song, and J. Sun. Foreground estimation based on linear regression model with fused sparsity on outliers. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(8):1346–1357, Aug 2013.

[26] T. Zhou and D. Tao. Godec: Randomized low-rank & sparse matrix decomposition in noisy case. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 33–40, 2011.

[27] X. Zhou, C. Yang, and W. Yu. Moving object detection by detecting contiguous outliers in the low-rank representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(3):597–610, 2013.