Moving Object Detection by Detecting Contiguous Outliers in the Low-Rank Representation

Xiaowei Zhou, *Student Member*, *IEEE*, Can Yang, and Weichuan Yu, *Member*, *IEEE*

Abstract—Object detection is a fundamental step for automated video analysis in many vision applications. Object detection in a video is usually performed by object detectors or background subtraction techniques. Often, an object detector requires manually labeled examples to train a binary classifier, while background subtraction needs a training sequence that contains no objects to build a background model. To automate the analysis, object detection without a separate training phase becomes a critical task. People have tried to tackle this task by using motion information. But existing motion-based methods are usually limited when coping with complex scenarios such as nonrigid motion and dynamic background. In this paper, we show that the above challenges can be addressed in a unified framework named DEtecting Contiguous Outliers in the LOw-rank Representation (DECOLOR). This formulation integrates object detection and background learning into a single process of optimization, which can be solved by an alternating algorithm efficiently. We explain the relations between DECOLOR and other sparsity-based methods. Experiments on both simulated data and real sequences demonstrate that DECOLOR outperforms the state-of-the-art approaches and it can work effectively on a wide range of complex scenarios.

Index Terms—Moving object detection, low-rank modeling, Markov Random Fields, motion segmentation

1 Introduction

Automated video analysis is important for many vision applications, such as surveillance, traffic monitoring, augmented reality, vehicle navigation, etc. [1], [2]. As pointed out in [1], there are three key steps for automated video analysis: object detection, object tracking, and behavior recognition. As the first step, object detection aims to locate and segment interesting objects in a video. Then, such objects can be tracked from frame to frame, and the tracks can be analyzed to recognize object behavior. Thus, object detection plays a critical role in practical applications.

Object detection is usually achieved by object detectors or background subtraction [1]. An object detector is often a classifier that scans the image by a sliding window and labels each subimage defined by the window as either object or background. Generally, the classifier is built by offline learning on separate datasets [3], [4] or by online learning initialized with a manually labeled frame at the start of a video [5], [6]. Alternatively, background subtraction [7] compares images with a background model and detects the changes as objects. It usually assumes that no object appears in images when building the background model [8], [2]. Such requirements of training examples for object or background modeling actually limit the applicability of above-mentioned methods in automated video analysis.

Another category of object detection methods that can avoid training phases are motion-based methods [1], [2], which only use motion information to separate objects from the background. The problem can be rephrased as follows: *Given a sequence of images in which foreground objects are present and moving differently from the background, can we separate the objects from the background automatically?* Fig. 1a shows such an example, where a walking lady is always present and recorded by a handheld camera. The goal is to take the image sequence as input and directly output a mask sequence of the walking lady.

The most natural way for motion-based object detection is to classify pixels according to motion patterns, which is usually named motion segmentation [9], [10]. These approaches achieve both segmentation and optical flow computation accurately and they can work in the presence of large camera motion. However, they assume rigid motion [9] or smooth motion [10] in respective regions, which is not generally true in practice. In practice, the foreground motion can be very complicated with nonrigid shape changes. Also, the background may be complex, including illumination changes and varying textures such as waving trees and sea waves. Fig. 1b shows such a challenging example. The video includes an operating escalator, but it should be regarded as background for human tracking purpose. An alternative motion-based approach is background estimation [11], [12]. Different from background subtraction, it estimates a background model directly from the testing sequence. Generally, it tries to seek temporal intervals inside which the pixel intensity is unchanged and uses image data from such intervals for background estimation. However, this approach also relies on the assumption of static background. Hence, it is difficult to handle the scenarios with complex background or moving cameras.

- *The authors are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. E-mail: {eexwzhou, eeyang, eeyu}@ust.hk.*

Manuscript received 3 Aug. 2011; revised 27 Mar. 2012; accepted 30 May 2012; published online 8 June 2012.
Recommended for acceptance by D. Cremers.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2011-08-0524.
Digital Object Identifier no. 10.1109/TPAMI.2012.132.

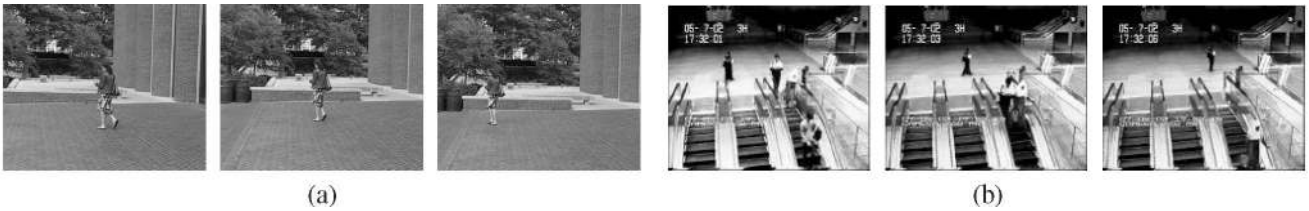


Fig. 1. Two examples to illustrate the problem. (a) A sequence of 40 frames, where a walking lady is recorded by a handheld camera. From left to right are the first, 20th, and 40th frames. (b) A sequence of 48 frames clipped from a surveillance video at the airport. From left to right are the first, 24th, and 48th frames. Notice that the escalator is moving. The objective is to segment the moving people automatically without extra inputs.

In this paper, we propose a novel algorithm for moving object detection which falls into the category of motion-based methods. It solves the challenges mentioned above in a unified framework named DEtecting Contiguous Outliers in the LOw-rank Representation (DECOLOR). We assume that the underlying background images are linearly correlated. Thus, the matrix composed of vectorized video frames can be approximated by a low-rank matrix, and the moving objects can be detected as outliers in this low-rank representation. Formulating the problem as outlier detection allows us to get rid of many assumptions on the behavior of foreground. The low-rank representation of background makes it flexible to accommodate the global variations in the background. Moreover, DECOLOR performs object detection and background estimation simultaneously without training sequences. The main contributions can be summarized as follows:

1. We propose a new formulation of outlier detection in the low-rank representation in which the outlier support and the low-rank matrix are estimated simultaneously. We establish the link between our model and other relevant models in the framework of Robust Principal Component Analysis (RPCA) [13]. Differently from other formulations of RPCA, we model the outlier support explicitly. DECOLOR can be interpreted as ℓ_0-penalty regularized RPCA, which is a more faithful model for the problem of moving object segmentation. Following the novel formulation, an effective and efficient algorithm is developed to solve the problem. We demonstrate that, although the energy is nonconvex, DECOLOR achieves better accuracy in terms of both object detection and background estimation compared against the state-of-the-art algorithm of RPCA [13].
2. In other models of RPCA, no prior knowledge on the spatial distribution of outliers has been considered. In real videos, the foreground objects usually are small clusters. Thus, contiguous regions should be preferred to be detected. Since the outlier support is modeled explicitly in our formulation, we can naturally incorporate such contiguity prior using Markov Random Fields (MRFs) [14].
3. We use a parametric motion model to compensate for camera motion. The compensation of camera motion is integrated into our unified framework and computed in a batch manner for all frames during segmentation and background estimation.

The MATLAB implementation of DECOLOR, experimental data, and more results are publicly available at http://bioinformatics.ust.hk/decolor/decolor.html.

2 Related Work

Previous methods for object detection are vast, including object detectors (supervised learning), image segmentation, background subtraction, etc., [1]. Our method aims to segment objects based on motion information and it comprises a component of background modeling. Thus, motion segmentation and background subtraction are the most related topics to this paper.

2.1 Motion Segmentation

In motion segmentation, the moving objects are continuously present in the scene, and the background may also move due to camera motion. The target is to separate different motions.

A common approach for motion segmentation is to partition the dense optical-flow field [15]. This is usually achieved by decomposing the image into different motion layers [16], [17], [10]. The assumption is that the optical-flow field should be smooth in each motion layer, and sharp motion changes only occur at layer boundaries. Dense optical flow and motion boundaries are computed in an alternating manner named *motion competition* [10], which is usually implemented in a level set framework. A similar scheme is later applied to dynamic texture segmentation [18], [19], [20]. While high accuracy can be achieved in these methods, accurate motion analysis itself is a challenging task due to the difficulties raised by aperture problem, occlusion, video noises, etc. [21]. Moreover, most of the motion segmentation methods require object contours to be initialized and the number of foreground objects to be specified [10].

An alternative approach for motion segmentation tries to segment the objects by analyzing point trajectories [9], [22], [23], [24]. Some sparse feature points are first detected and tracked throughout the video and then separated into several clusters via subspace clustering [25] or spectral clustering [24]. The formulation is mathematically elegant and it can handle large camera motion. However, these methods require point trajectories as input and only output a segmentation of sparse points. The performance relies on the quality of point tracking and postprocessing is needed to obtain the dense segmentation [26]. Also, they are limited when dealing with noisy data and nonrigid motion [25].

2.2 Background Subtraction

In background subtraction, the general assumption is that a background model can be obtained from a training sequence that does not contain foreground objects. Moreover, it usually assumes that the video is captured by a static camera [7]. Thus, foreground objects can be detected by checking the difference between the testing frame and the background model built previously.

A considerable number of works have been done on background modeling, i.e., building a proper representation of the background scene. Typical methods include single Gaussian distribution [27], Mixture of Gaussian (MoG) [28], kernel density estimation [29], [30], block correlation [31], codebook model [32], Hidden Markov model [33], [34], and linear autoregressive models [8], [35], [36].

Learning with sparsity has drawn a lot of attention in recent machine learning and computer vision research [37], and several methods based on the sparse representation for background modeling have been developed. One pioneering work is the *eigen backgrounds* model [38], where the principal component analysis (PCA) is performed on a training sequence. When a new frame arrives, it is projected onto the subspace spanned by the principal components, and the residues indicate the presence of new objects. An alternative approach that can operate sequentially is sparse signal recovery [39], [40], [41]. Background subtraction is formulated as a regression problem with the assumption that a new-coming frame should be sparsely represented by a linear combination of preceding frames except for foreground parts. These models capture the correlation between video frames. Thus, they can naturally handle global variations in the background such as illumination change and dynamic textures.

Background subtraction methods mentioned above rarely consider the scenario where the objects appear at the start and are continuously present in the scene (i.e., the training sequence is not available). Very little literature considers the problem of background initialization [11], [42]. Most of them seek a stable interval, inside which the intensity is relatively smooth for each pixel independently. Pixels during such intervals are regarded as background, and the background scene is estimated from these intervals. The validity of this approach relies on the assumption of static background. Thus, it is limited when processing dynamic background or videos captured by a moving camera.

3 Contiguous Outlier Detection in the Low-Rank Representation

In this section, we focus on the problem of detecting contiguous outliers in the low-rank representation. We first consider the case without camera motion. We will discuss the scenarios with moving cameras in Section 4.

3.1 Notations

In this paper, we use following notations. $I_j \in \mathbb{R}^m$ denotes the jth frame of a video sequence, which is written as a column vector consisting of m pixels. The ith pixel in the jth frame is denoted as ij. $D = [I_1, \dots, I_n] \in \mathbb{R}^{m\times n}$ is a matrix representing all n frames of a sequence. $B \in \mathbb{R}^{m\times n}$ is a matrix with the same size of D, which denotes the underlying background images. $S \in \{0,1\}^{m\times n}$ is a binary matrix denoting the foreground support:

$$S_{ij} = \begin{cases} 0, & \text{if } ij \text{ is background} \\ 1, & \text{if } ij \text{ is foreground.} \end{cases} \tag{1}$$

We use $\mathcal{P}_S(X)$ to represent the orthogonal projection of a matrix X onto the linear space of matrices supported by S,

$$\mathcal{P}_S(X)(i,j) = \begin{cases} 0, & \text{if } S_{ij} = 0 \\ X_{ij}, & \text{if } S_{ij} = 1, \end{cases} \tag{2}$$

and $\mathcal{P}_{S^\perp}(X)$ to be its complementary projection, i.e., $\mathcal{P}_S(X) + \mathcal{P}_{S^\perp}(X) = X$.

Four norms of a matrix are used throughout this paper. $\|X\|_0$ denotes the ℓ_0-norm, which counts the number of nonzero entries. $\|X\|_1 = \sum_{ij}|X_{ij}|$ denotes the ℓ_1-norm. $\|X\|_F = \sqrt{\sum_{ij} X_{ij}^2}$ is the Frobenius norm. $\|X\|_*$ means the nuclear norm, i.e., sum of singular values.

3.2 Formulation

Given a sequence D, our objective is to estimate the foreground support S as well as the underlying background images B. To make the problem well posed, we have the following models to describe the foreground, the background, and the formation of observed signal.

Background model. The background intensity should be unchanged over the sequence except for variations arising from illumination change or periodical motion of dynamic textures.[1] Thus, background images are linearly correlated with each other, forming a low-rank matrix B. Besides the low-rank property, we don't make any additional assumption on the background scene. Thus, we only impose the following constraint on B:

$$\operatorname{rank}(B) \le K, \tag{3}$$

where K is a constant to be predefined. Intrinsically, K constrains the complexity of the background model. We will discuss more on this parameter in Section 5.1.

Foreground model. The foreground is defined as any object that moves differently from the background. Foreground motion gives intensity changes that cannot be fitted into the low-rank model of background. Thus, they can be detected as outliers in the low-rank representation. Generally, we have a prior that foreground objects should be contiguous pieces with relatively small size. The binary states of entries in foreground support S can be naturally modeled by a Markov Random Field [43], [14]. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices denoting all $m \times n$ pixels in the sequence and E is the set of edges connecting spatially or temporally neighboring pixels. Then, the energy of S is given by the Ising model [14]:

$$\sum_{ij\in\mathcal{V}} u_{ij}(S_{ij}) + \sum_{(ij,kl)\in\mathcal{E}} \lambda_{ij,kl}|S_{ij} - S_{kl}|, \tag{4}$$

where u_{ij} denotes the unary potential of S_{ij} being 0 or 1, and the parameter $\lambda_{ij,kl} > 0$ controls the strength of dependency between S_{ij} and S_{kl}. To prefer $S_{ij} = 0$ that indicates sparse foreground, we define the unary potential u_{ij} as

$$u_{ij}(S_{ij}) = \begin{cases} 0, & \text{if } S_{ij} = 0 \\ \lambda_{ij}, & \text{if } S_{ij} = 1, \end{cases} \tag{5}$$

where the parameter $\lambda_{ij} > 0$ penalizes $S_{ij} = 1$. For simplicity, we set λ_{ij} and $\lambda_{ij,kl}$ as constants over all locations. That

[1] Background motion caused by moving cameras will be considered in Section 4.

is, $\lambda_{ij} = \beta$ and $\lambda_{ij,kl} = \gamma$, where $\beta > 0$ and $\gamma > 0$ are positive constants. This means that we have no additional prior about the locations of objects.

Signal model. The signal model describes the formation of D, given B and S. In the background region where $S_{ij} = 0$, we assume that $D_{ij} = B_{ij} + \epsilon_{ij}$, where ϵ_{ij} denotes i.i.d. Gaussian noise. That is, $D_{ij} \sim \mathcal{N}(B_{ij}, \sigma^2)$, with σ^2 being the variance of Gaussian noise. Thus, B_{ij} should be the best fitting to D_{ij} in the least squares sense when $S_{ij} = 0$. In the foreground regions where $S_{ij} = 1$, the background scene is occluded by the foreground. Thus, D_{ij} equals the foreground intensity. Since we don't make any assumption about the foreground appearance, D_{ij} is not constrained when $S_{ij} = 1$.

Combining above three models, we propose to minimize the following energy to estimate B and S:

$$\begin{aligned} \min_{B, S_{ij} \in \{0,1\}} \quad & \frac{1}{2} \sum_{ij: S_{ij}=0} (D_{ij} - B_{ij})^2 \\ & + \beta \sum_{ij} S_{ij} + \gamma \sum_{(ij,kl) \in \mathcal{E}} |S_{ij} - S_{kl}|, \\ \text{s.t.} \quad & \operatorname{rank}(B) \leq K. \end{aligned} \tag{6}$$

This formulation says that the background images should form a low-rank matrix and fit the observed sequence in the least squares sense except for foreground regions that are sparse and contiguous.

To make the energy minimization tractable, we relax the rank operator on B with the nuclear norm. The nuclear norm has proven to be an effective convex surrogate of the rank operator [44]. Moreover, it can help to avoid overfitting, which will be illustrated by experiments in Section 5.1.2.

Writing (6) in its dual form and introducing matrix operators, we obtain the final form of the energy function:

$$\begin{aligned} \min_{B, S_{ij} \in \{0,1\}} & \frac{1}{2} \| \mathcal{P}_{S^\perp}(D - B) \|_F^2 + \alpha \|B\|_* \\ & + \beta \|S\|_1 + \gamma \, \|A \text{vec}(S)\|_1. \end{aligned} \tag{7}$$

Here, A is the node-edge incidence matrix of $\mathcal{G}$, and $\alpha > 0$ is a parameter associated with K, which controls the complexity of the background model. Proper choice of α, β, and γ will be discussed in detail in Section 3.3.3.

3.3 Algorithm

The objective function defined in (7) is nonconvex and it includes both continuous and discrete variables. Joint optimization over B and S is extremely difficult. Hence, we adopt an alternating algorithm that separates the energy minimization over B and S into two steps. B-step is a convex optimization problem and S-step is a combinatorial optimization problem. It turns out that the optimal solutions of B-step and S-step can be computed efficiently.

3.3.1 Estimation of the Low-Rank Matrix B

Given an estimate of the support $\hat{S}$, the minimization in (7) over B turns out to be the matrix completion problem [45]:

$$\min_B \frac{1}{2} \| \mathcal{P}_{\hat{S}^\perp}(D - B) \|_F^2 + \alpha \|B\|_*. \tag{8}$$

This is to learn a low-rank matrix from partial observations. The optimal B in (8) can be computed efficiently by the SOFT-IMPUTE algorithm [45], which makes use of the following lemma [46]:

Lemma 1. *Given a matrix Z, the solution to the optimization problem*

$$\min_X \frac{1}{2} \|Z - X\|_F^2 + \alpha \|X\|_* \tag{9}$$

is given by $\hat{X} = \Theta_\alpha(Z)$, where Θ_α means the singular value thresholding

$$\Theta_\alpha(Z) = U \Sigma_\alpha V^T. \tag{10}$$

Here, $\Sigma_\alpha = \operatorname{diag}[(d_1 - \alpha)_+, \ldots, (d_r - \alpha)_+]$, $U \Sigma V^T$ is the SVD of Z, $\Sigma = \operatorname{diag}[d_1, \ldots, d_r]$, and $t_+ = \max(t, 0)$.

Rewriting (8), we have

$$\begin{aligned} & \min_B \frac{1}{2} \| \mathcal{P}_{\hat{S}^\perp}(D - B) \|_F^2 + \alpha \|B\|_* \\ & = \min_B \frac{1}{2} \| [\mathcal{P}_{\hat{S}^\perp}(D) + \mathcal{P}_{\hat{S}}(B)] - B \|_F^2 + \alpha \|B\|_*. \end{aligned} \tag{11}$$

Using Lemma 1, the optimal solution to (8) can be obtained by iteratively using

$$\hat{B} \leftarrow \Theta_\alpha(\mathcal{P}_{\hat{S}^\perp}(D) + \mathcal{P}_{\hat{S}}(\hat{B})) \tag{12}$$

with arbitrarily initialized $\hat{B}$. Please refer to [45] for the details of SOFT-IMPUTE and the proof of its convergence.

3.3.2 Estimation of the Outlier Support S

Next, we investigate how to minimize the energy in (7) over S given the low-rank matrix $\hat{B}$. Noticing that $S_{ij} \in \{0, 1\}$, the energy can be rewritten as follows:

$$\begin{aligned} & \frac{1}{2} \| \mathcal{P}_{S^\perp}(D - \hat{B}) \|_F^2 + \beta \|S\|_1 + \gamma \|A \text{ vec}(S)\|_1 \\ & = \frac{1}{2} \sum_{ij} (D_{ij} - \hat{B}_{ij})^2 (1 - S_{ij}) + \beta \sum_{ij} S_{ij} + \gamma \|A \text{ vec}(S)\|_1 \\ & = \sum_{ij} \left(\beta - \frac{1}{2} (D_{ij} - \hat{B}_{ij})^2 \right) S_{ij} + \gamma \|A \text{ vec}(S)\|_1 + \mathcal{C}, \end{aligned} \tag{13}$$

where $\mathcal{C} = \frac{1}{2} \sum_{ij} (D_{ij} - \hat{B}_{ij})^2$ is a constant when $\hat{B}$ is fixed. The above energy is in the standard form of the first-order MRFs with binary labels, which can be solved exactly using graph cuts [47], [48].

Ideally, both spatial and temporal smoothness can be imposed by connecting all pairs of nodes in $\mathcal{G}$ which correspond to all pairs of spatially or temporally neighboring pixels in the sequence. However, this will make $\mathcal{G}$ extremely large and difficult to solve. In implementation, we only connect spatial neighbors. Thus, $\mathcal{G}$ can be separated into subgraphs of single images, and the graph cuts can be operated for each image separately. This dramatically reduces the computational cost. Based on our observation, the spatial smoothness is sufficient to obtain satisfactory results.

3.3.3 Parameter Tuning

The parameter α in (7) controls the complexity of the background model. A larger α gives a $\hat{B}$ with smaller nuclear norm. In our algorithm, we first give a rough estimate to the rank of the background model, i.e., K in (6). Then, we start from a large α. After each run of SOFT-IMPUTE, if $\mathrm{rank}(\hat{B}) \leq K$, we reduce α by a factor $\eta_1 < 1$ and repeat SOFT-IMPUTE until $\mathrm{rank}(\hat{B}) > K$. Using *warm-start*, this sequential optimization is efficient [45]. In our implementation, we initialize α to be the second largest singular value of D, and $\eta_1 = 1/\sqrt{2}$.

The parameter β in (7) controls the sparsity of the outlier support. From (13), we can see that $\hat{S}_{ij}$ is more likely to be 1 if $\frac{1}{2}(D_{ij} - \hat{B}_{ij})^2 > \beta$. Thus, the choice of β should depend on the noise level in images. Typically, we set $\beta = 4.5\hat{\sigma}^2$, where $\hat{\sigma}^2$ is estimated online by the variance of $D_{ij} - \hat{B}_{ij}$. Since the estimation of $\hat{B}$ and $\hat{\sigma}$ is biased at the beginning iterations, we propose to start our algorithm with a relatively large β, and then reduce β by a factor $\eta_2 = 0.5$ after each iteration until β reaches $4.5\hat{\sigma}^2$. In other words, we tolerate more error in model fitting at the beginning since the model itself is not accurate enough. With the model estimation getting better and better, we decrease the threshold and declare more and more outliers.

In conclusion, we only have two parameters to choose, i.e., K and γ. In Section 5.1.2, we will show that DECOLOR performs stably if K and γ are in proper ranges. In all our experiments, we let $K = \sqrt{n}$ and $\gamma = \beta$ and 5β for simulation and real sequences, respectively.

3.3.4 Convergence

For fixed parameters, we always minimize a single lower bounded energy in each step. The convergence property of SOFT-IMPUTE has been proven in [45]. Therefore, the algorithm must converge to a local minimum. For adaptive parameter tuning, our strategy guarantees that the coefficients (α, β, γ) keep decreasing for each change. Thus, the energy in (7) decreases monotonically with the algorithm running. Furthermore, we can manually set lower bounds for both α and β to stop the iteration. Empirically, DECOLOR converges in about 20 iterations for a convergence precision of 10^{-5}.

3.4 Relation to Other Methods

3.4.1 Robust Principal Component Analysis

RPCA has drawn a lot of attention in computer vision [49], [50]. Recently, the seminal work [13] showed that, under some mild conditions, the low-rank model can be recovered from unknown corruption patterns via a convex program named Principal Component Pursuit (PCP). The examples in [13] demonstrate the superior performance of PCP compared with previous methods of RPCA and its promising potential for background subtraction.

As discussed in [13], PCP can be regarded as a special case of the following decomposition model:

$$D = B + E + \epsilon, \tag{14}$$

where B is a low-rank matrix, E represents the intensity shift caused by outliers, and ϵ denotes the Gaussian noise. PCP only seeks for the low-rank and sparse decomposition $D = B + E$ without considering ϵ. Recently, Stable Principal Component Pursuit (SPCP) has been proposed [51]. It extends PCP [13] to handle both sparse gross errors and small entrywise noises. It tries to find the decomposition by minimizing the following energy:

$$\min_{B,E} \frac{1}{2}\|D - B - E\|_F^2 + \alpha\,\mathrm{rank}(B) + \beta\|E\|_0. \tag{15}$$

To make the optimization tractable, (15) is relaxed by replacing $\mathrm{rank}(B)$ with $\|B\|_*$ and $\|E\|_0$ with $\|E\|_1$ in PCP or SPCP. Thus, the problem turns out to be convex and can be solved efficiently via convex optimization. However, the ℓ_1 relaxation requires that the distribution of corruption should be sparse and random enough, which is not generally true in the problem of motion segmentation. Experiments in Section 5 show that PCP is not robust enough when the moving objects take up relatively large and contiguous space of the sequence.

Next, we shall explain the relation between our formulation in (7) and the formulation in (15). It is easy to see that, as long as $E_{ij} \neq 0$, we must have $E_{ij} = D_{ij} - B_{ij}$ to minimize (15). Thus, (15) has the same minimizer with the following energy:

$$\min_{B,E} \frac{1}{2}\sum_{ij:E_{ij}=0} (D_{ij} - B_{ij})^2 + \alpha\,\mathrm{rank}(B) + \beta\|E\|_0. \tag{16}$$

The first term in (16) can be rewritten as $\frac{1}{2}\|\mathcal{P}_{S^\perp}(D - B)\|_F^2$. Noticing that $\|E\|_0 = \|S\|_1$ and replacing $\mathrm{rank}(B)$ with $\|B\|_*$, (16) can be finally rewritten as (7) if the last smoothness term in (7) is ignored.

Thus, DECOLOR can be regarded as a special form of RPCA where the ℓ_0-penalty on E is not relaxed and the problem in (15) is converted to the optimization over S in (6). One recent work [52] has shown that the ℓ_0-penalty works effectively for outlier detection in regression, while the ℓ_1-penalty does not. As pointed out in [52], the theoretical reason for the unsatisfactory performance of the ℓ_1-penalty is that the irrepresentable condition [53] is often not satisfied in the outlier detection problem. In order to go beyond the ℓ_1-penalty, nonconvex penalties have been explored in recent literature [52], [54]. Compared with the ℓ_1-norm, nonconvex penalties give an estimation with less bias but higher variance. Thus, these nonconvex penalties are superior to the ℓ_1-penalty when the signal-noise-ratio (SNR) is relatively high [54]. For natural video analysis, it is the case.

In summary, both PCP [13] and DECOLOR aim to recover a low-rank model from corrupted data. PCP [13], [51] uses the convex relaxation by replacing $\mathrm{rank}(B)$ with $\|B\|_*$ and $\|E\|_0$ with $\|E\|_1$. DECOLOR only relaxes the rank penalty and keeps the ℓ_0-penalty on E to preserve the robustness to outliers. Moreover, DECOLOR estimates the outlier support S explicitly by formulating the problem as the energy minimization over S, and models the continuity prior on S using MRFs to improve the accuracy of detecting contiguous outliers.

3.4.2 Sparse Signal Recovery

With the success of compressive sensing [55], sparse signal recovery has become a popular framework to deal with various problems in machine learning and signal processing [37], [56], [57]. To make use of structural information about nonzero patterns of variables, the structured sparsity is defined in recent works [58], [59], and several algorithms have been developed and applied successfully on background subtraction, such as Lattice Matching Pursuit (LaMP) [39], Dynamic Group Sparsity (DGS) recovery [40], and Proximal Operator using Network Flow (ProxFlow) [41].

In sparse signal recovery for background subtraction, a testing image $y \in \mathbb{R}^m$ is modeled as a sparse linear combination of n previous frames $\Phi \in \mathbb{R}^{m\times n}$ plus a sparse error term $e \in \mathbb{R}^m$ and a Gaussian noise term $\epsilon \in \mathbb{R}^m$:

$$y = \Phi w + e + \epsilon. \tag{17}$$

$w \in \mathbb{R}^n$ is the coefficient vector. The first term Φw accounts for the background shared between y and Φ, while the sparse error e corresponds to the foreground in y. Thus, background subtraction can be achieved by recovering w and e. Taking the latest algorithm ProxFlow [41] as an example, the following optimization is proposed:

$$\min_{w,e} \frac{1}{2}\|y - \Phi w - e\|_2^2 + \lambda_1\|w\|_1 + \lambda_2\|e\|_{\ell_1/\ell_\infty}, \tag{18}$$

where $\|\cdot\|_{\ell_1/\ell_\infty}$ is a norm to induce the group-sparsity. Please refer to [41] for the detailed definition. In short, the ℓ_1/ℓ_∞-norm is used as a structured regularizer to encode the prior that nonzero entries of e should be in a group structure, where the groups are specified to be all overlapping 3×3-squares on the image plane [41].

In (17), Φ can be interpreted as a basis matrix for linear regression to fit the testing image y. In the literature mentioned above, Φ is fixed to be the training sequence [41] or previous frames on which background subtraction has been performed [40]. Then, the only task is to recover the sparse coefficients.

In our problem formulation, Φ is unknown. DECOLOR learns the bases and coefficients for a batch of test images simultaneously. To illustrate this, we can rewrite (14) as

$$D = \Phi W + E + \epsilon, \tag{19}$$

where the original low-rank B is factorized as a product of a basis matrix $\Phi \in \mathbb{R}^{m\times r}$ and a coefficient matrix $W \in \mathbb{R}^{r\times n}$ with r being the rank of B.

In summary, LaMP, DGS, and ProxFlow aim to detect new objects in a new testing image given a training sequence not containing such objects. The problem is formulated as linear regression with fixed bases. DECOLOR aims to segment moving objects from a short sequence during which the objects continuously appear, which is a more challenging problem. To this end, DECOLOR estimates the foreground and background jointly by outlier detection during matrix learning. The difference between DECOLOR and sparse signal recovery will be further demonstrated using experiments on real sequences in Section 5.2.1.

4 Extension to Moving Background

The above derivation is based on the assumption that the videos are captured by static cameras. In this section, we introduce domain transformations into our model to compensate for the background motion caused by moving cameras. Here, we use the 2D parametric transforms [60] to model the translation, rotation, and planar deformation of the background.

Let $D_j \circ \tau_j$ denote the jth frame after the transformation parameterized by vector $\tau_j \in \mathbb{R}^p$, where p is the number of parameters of the motion model (e.g., $p = 6$ for the affine motion or $p = 8$ for the projective motion). Then, the proposed decomposition becomes $D \circ \tau = B + E + \epsilon$, where $D \circ \tau = [D_1 \circ \tau_1, \ldots, D_n \circ \tau_n]$ and $\tau \in \mathbb{R}^{p\times n}$ is a vector comprising all τ_j. A similar idea can be found in the recent work on batch image alignment [57].

Next, we substitute D in (7) with $D \circ \tau$ and estimate τ along with B, S by iteratively minimizing

$$\begin{aligned}\min_{\tau,B,S} \frac{1}{2}\|\mathcal{P}_{S^\perp}(D\circ\tau - B)\|_F^2 + \alpha\|B\|_* \\ + \beta\|S\|_1 + \gamma\|A\mathrm{vec}(S)\|_1.\end{aligned} \tag{20}$$

Now, we investigate how to minimize the energy in (20) over τ, given $\hat{B}$ and $\hat{S}$:

$$\hat{\tau} = \arg\min_{\tau} \|\mathcal{P}_{\hat{S}^\perp}(D\circ\tau - \hat{B})\|_F^2. \tag{21}$$

Here, we use the incremental refinement [57], [60] to solve this parametric motion estimation problem: At each iteration, we update $\hat{\tau}$ by a small increment $\Delta\tau$ and linearize $D \circ \tau$ as $D \circ \hat{\tau} + J_{\hat{\tau}}\Delta\tau$, where $J_{\hat{\tau}}$ denotes the Jacobian matrix $\frac{\partial D}{\partial \tau}|_{\tau=\hat{\tau}}$. Thus, τ can be updated in the following way:

$$\hat{\tau} \leftarrow \hat{\tau} + \arg\min_{\Delta\tau} \|\mathcal{P}_{\hat{S}^\perp}(D\circ\hat{\tau} - \hat{B} + J_{\hat{\tau}}\Delta\tau)\|_F^2. \tag{22}$$

The minimization over $\Delta\tau$ in (22) is a weighted least squares problem which has a closed-form solution.

In practice, the update of $\tau_1, \ldots, \tau_n$ can be done separately since the transformation is applied on each image individually. Thus, the update of τ is efficient. To accelerate the convergence of DECOLOR, we initialize τ by roughly aligning each frame D_j to the middle frame $D_{\frac{n}{2}}$ before the main loops of DECOLOR. The prealignment is done by the robust multiresolution method proposed in [61].

All steps of DECOLOR with adaptive parameter tuning are summarized in Algorithm 1.

Algorithm 1. Moving Object Segmentation by DECOLOR

```
1.  Input: $D = [I_1, \ldots, I_n] \in \mathbb{R}^{m\times n}$
2.  Initialize: $\hat{\tau}, \hat{B} \leftarrow D\circ\hat{\tau}, \hat{S} \leftarrow \mathbf{0}, \alpha, \beta$.
3.  repeat
4.    $\hat{\tau} \leftarrow \hat{\tau} + \mathrm{argmin}_{\Delta\tau}\|\mathcal{P}_{\hat{S}^\perp}(D\circ\hat{\tau} - \hat{B} + J_{\hat{\tau}}\Delta\tau)\|_2^2$;
5.    repeat
6.      $\hat{B} \leftarrow \Theta_\alpha(\mathcal{P}_{\hat{S}^\perp}(D\circ\hat{\tau}) + \mathcal{P}_{\hat{S}}(\hat{B}))$;
7.    until convergence
8.    if $\mathrm{rank}(\hat{B}) \leq K$ then
9.      $\alpha \leftarrow \eta_1\alpha$;
10.     go to Step 5;
11.   end if
12.   estimate $\hat{\sigma}$;
13.   $\beta \leftarrow \max(\eta_2\beta, 4.5\hat{\sigma}^2)$;
14.   $\hat{S} \leftarrow \arg\min_S \sum_{ij}(\beta - \frac{1}{2}([D\circ\hat{\tau}]_{ij} - \hat{B}_{ij})^2)S_{ij}$
        $+\gamma\|A\ \mathrm{vec}(S)\|_1$
15. until convergence
16. Output: $\hat{B}$,$\hat{S}$,$\hat{\tau}$
```

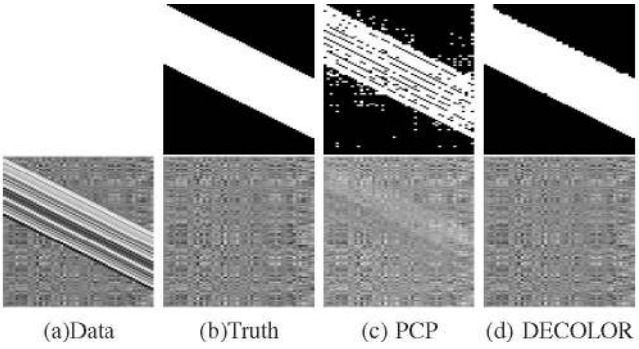


Fig. 2. (a) An example of synthesized data. Sequence $D \in \mathbb{R}^{100\times 50}$ is a matrix composed of 50 frames of 1D images with 100 pixels per image. (b) The foreground support S_0 and underlying background images B_0. $\mathrm{rank}(B_0) = 3$. D is generated by adding a foreground object with width $W = 40$ to each column of B_0, which moves downward for 1 pixel per column. Also, i.i.d. Gaussian noise is added to each entry, and $\mathrm{SNR} = 10$. (c) The results of PCP. The top panel is $\hat{S}$ and the bottom panel is $\hat{B}$. $\hat{S}$ of PCP is obtained by thresholding $|D_{ij} - \hat{B}_{ij}|$ with a threshold that gives the largest F-measure. Notice the artifacts in both $\hat{S}$ and $\hat{B}$ estimated by PCP. (d) The results of DECOLOR. Here, $\hat{S}$ is directly output by DECOLOR without postprocessing.

5 Experiments

5.1 Simulation

In this section, we perform numerical experiments on synthesized data. We consider the situations with no background motion and mainly investigate whether DECOLOR can successfully separate the contiguous outliers from the low-rank model.

To better visualize the data, we use a simplified scenario: The video to be segmented is composed of 1D images. Thus, the image sequence and results can be displayed as 2D matrices. We generate the input D by adding a foreground occlusion with support S_0 to a background matrix B_0. The background matrix B_0 with rank r is generated as $B_0 = UV^T$, where U and V are $m \times r$ and $n \times r$ matrices with entries independently sampled from a standard normal distribution. We choose $m = 100$, $n = 50$, and $r = 3$ for all experiments. Then, an object with width W is superimposed on each column of B_0 and shifts downward for 1 pixel per column. The intensity of this object is independently sampled from a uniform distribution $\mathcal{U}(-c, c)$, where c is chosen to be the largest magnitude of entries in B_0. Also, we add i.i.d. Gaussian noise ϵ to D with the corresponding signal-to-noise ratio defined as

$$\mathrm{SNR} = \sqrt{\frac{\mathrm{var(B_0)}}{\mathrm{var}(\epsilon)}}. \tag{23}$$

Fig. 2a shows an example where the moving foreground can be recognized as contiguous outliers superposed on a low-rank matrix. Our goal is to estimate S_0 and recover B_0 at the same time.

For quantitative evaluation, we measure the accuracy of outlier detection by comparing $\hat{S}$ with S_0. We regard it as a classification problem and evaluate the results using precision and recall, which are defined as

$$\mathrm{precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \quad \mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \tag{24}$$

where TP, FP, TN, and FN mean the numbers of true positives, false positives, true negatives, and false negatives, respectively. Precision and recall are widely used when the class distribution is skewed [62]. For simplicity, instead of plotting precision/recall curves, we use a single measurement named F-measure that combines precision and recall:

$$\text{F-measure} = 2\frac{\mathrm{precision}\cdot\mathrm{recall}}{\mathrm{precision}+\mathrm{recall}}. \tag{25}$$

The higher the F-measure is, the better the detection accuracy is. On our observation, PCP requires proper thresholding to generate a really sparse $\hat{S}$. For fair comparison, $\hat{S}$ of PCP is obtained by thresholding $|D_{ij} - \hat{B}_{ij}|$ with a threshold that gives the maximal F-measure. Furthermore, we measure the accuracy of low-rank recovery by calculating the difference between $\hat{B}$ and B_0. We use the Root Mean Square Error (RMSE) to measure the difference

$$RMSE = \frac{\|\hat{B} - B_0\|_F}{\|B_0\|_F}. \tag{26}$$

5.1.1 Comparison to PCP

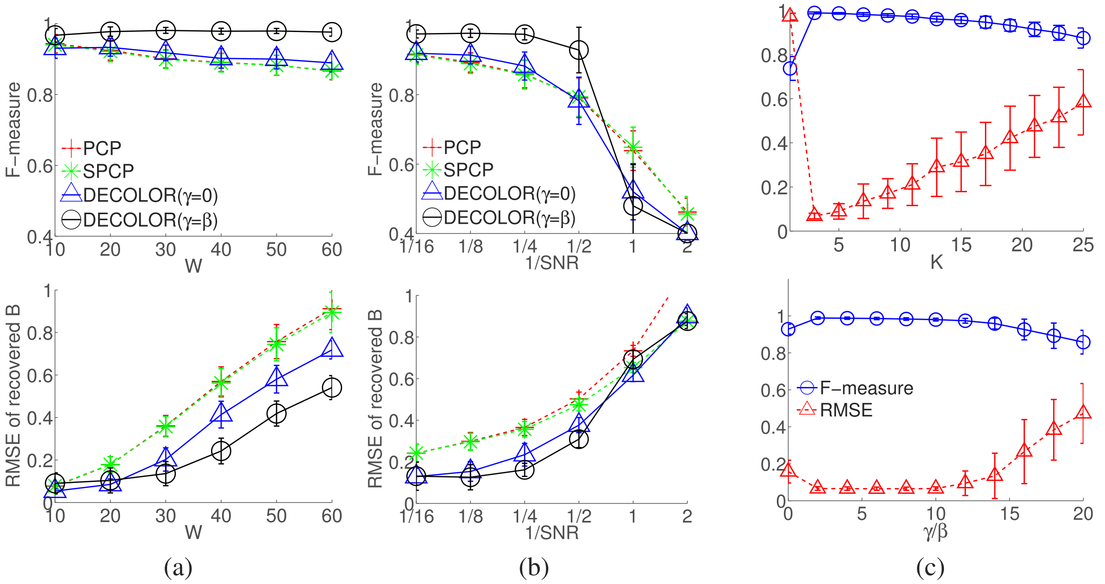
Fig. 2 gives a qualitative comparison between PCP and DECOLOR. Fig. 2c presents the results of PCP. Notice the artifacts in $\hat{B}$ that spatially coincide with S_0, which shows that the ℓ_1-penalty is not robust enough for relatively dense errors distributed in a contiguous region. Fig. 2d shows the results of DECOLOR. We see fewer false detections in estimated $\hat{S}$ compared with PCP. Also, the recovered $\hat{B}$ is less corrupted by outliers.

For quantitative evaluation, we perform random experiments with different object width W and SNR. Fig. 3a reports the numerical results as functions of W. We can see that all methods achieve a high accuracy when $W = 10$, which means all of them work well when outliers are really sparse. As W increases, the performance of PCP degrades significantly, while that of DECOLOR keeps less affected. This demonstrates the robustness of DECOLOR. The result of DECOLOR with $\gamma = 0$ falls in between those of PCP and DECOLOR with $\gamma = \beta$, and it has a larger variance. This shows the importance of the contiguity prior. Moreover, we can find that DECOLOR gives a very stable performance for outlier detection (F-measure), while the accuracy of matrix recovery (inverse to RMSE) drops obviously as W increases. The reason is that some background pixels are always occluded when the foreground is too large such that they cannot be recovered even when the foreground can be detected accurately.

Fig. 3b shows the results under different noise levels. DECOLOR maintains better performance than PCP if SNR is relatively high, but drops dramatically after $\mathrm{SNR} < 2$. This can be interpreted by the property of nonconvex penalties. Compared with ℓ_1-norm, nonconvex penalties are more robust to gross errors [63] but more sensitive to entrywise perturbations [54]. In general cases of natural video analysis, SNR is much larger than 1. Thus, DECOLOR can work stably.

5.1.2 Effects of Parameters

Fig. 3c demonstrates the effects of parameters in Algorithm 1, i.e., K and γ.



(a) (b) (c)

Fig. 3. Quantitative evaluation. (a) F-measure and RMSE as functions of W when $\mathrm{SNR} = 10$. (b) F-measure and RMSE as functions of SNR when $W = 25$. (c) The effects of parameters, i.e., K and γ. The results are averaged over 50 random trials with $W = 25$ and $SNR = 10$. The top panel shows the effect of K. The true rank of B_0 is 3. The accuracy increases sharply when K changes from 1 to 3 and decreases smoothly after K is larger than 3. The bottom panel shows the effect of γ. The accuracy keeps stable within $[\beta, 10\beta]$.

The parameter K is the rough estimate of $\mathrm{rank}(B_0)$, which controls the complexity of the background model. Here, the true rank of B_0 is 3. From the top plot in Fig. 3c, we can see that the optimal result is achieved at the turning point where $K = 3$. After that, the accuracy decreases very smoothly as K increases. This insensitivity to K is attributed to the shrinkage effect of the nuclear norm in (7), which plays an important role to prevent overfitting when estimating B. Specifically, given parameters K and α, the singular values of $\hat{B}$ are always shrunk by α due to the soft-thresholding operator in (10). Thus, our model overfits slowly when K is larger than the true rank. Similar results can be found in [45].

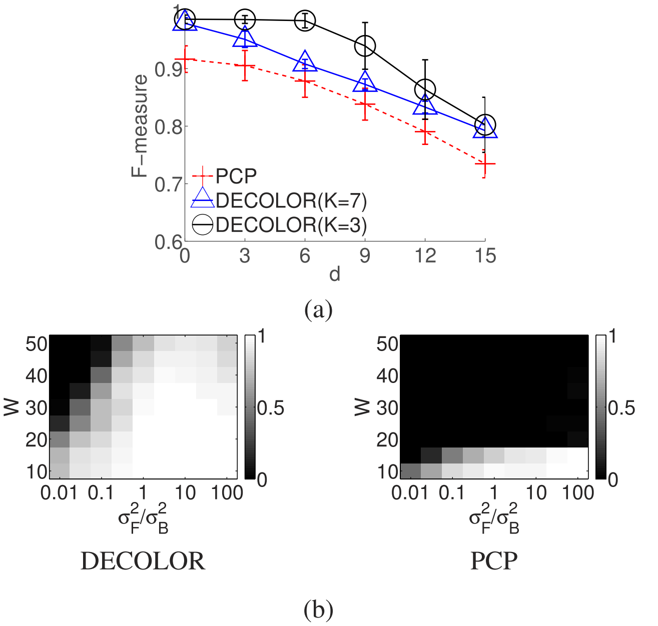
The parameter γ controls the strength of mutual interaction between neighboring pixels. From the bottom plot in Fig. 3c, we can see that the performance remains very stable when $\gamma \in [\beta, 10\beta]$.

5.1.3 Inseparable Cases

In previous simulations, the foreground was always moving and the foreground entries were sampled from a uniform distribution with a relatively large variance. Under these conditions, DECOLOR performs effectively and stably for foreground detection (F-measure) unless SNR is too bad. Next, we would like to study the cases when DECOLOR cannot separate the foreground from the background correctly.

First, we let the foreground not move for d frames when generating the data. Fig. 4a shows the averaged F-measure as a function of d. Here, $\mathrm{rank}(B_0) = 3$. We can see that, with the default parameter $K = 7$, the accuracy of DECOLOR will decrease dramatically as long as $d > 0$. This is because DECOLOR overfits the static foreground into the background model, as the model dimension K is larger than its actual value. When we decrease K to 3, DECOLOR performs more stably until $d > 6$, which means that DECOLOR can tolerate temporary stopping of foreground motion. In short, when the object is not always moving, DECOLOR becomes more sensitive to K, and it cannot work when the object stops for a long time.

Next, to investigate the influence of foreground texture, we also run DECOLOR on random problems with outlier entries sampled from uniform distributions with random mean and different variances σ_F^2. Fig. 4b displays the fraction of trials in which DECOLOR gives a high accuracy



(a)

DECOLOR PCP

(b)

Fig. 4. Simulation to illustrate inseparable cases of DECOLOR. (a) F-measure as a function of d, where d is the number of frames within which the foreground stops moving. The true rank of B_0 is 3. (b) Fraction of trials of accurate foreground detection ($\mathrm{F\text{-}measure} > 0.95$) over 200 trials as a function of σ_F and W. Here, σ_F represents the standard deviation of foreground intensities and W denotes the foreground width. σ_B is the standard deviation of B_0.

TABLE 1
Information of the Sequences Used in Experiments

Fig.	Size×#frames	Ref.	Description
Fig. 6(a)	$[160, 120] \times 48$	[42]	Crowded scene
Fig. 6(b)	$[238, 158] \times 24$	[18]	Crowded scene
Fig. 6(c)	$[160, 128] \times 24$	[64]	Crowded scene
Fig. 6(d)	$[160, 128] \times 48$	[64]	Dynamic background
Fig. 6(e)	$[160, 128] \times 48$	[64]	Dynamic background
Fig. 7(a)	$[320, 240] \times 40$	[24]	Moving cameras
Fig. 7(b)	$[320, 240] \times 30$	[24]	Moving cameras
Fig. 7(c)	$[320, 240] \times 30$	[24]	Moving cameras
Fig. 7(d)	$[320, 240] \times 24$	[24]	Moving cameras
Fig. 8	$[180, 144] \times 48$	[20]	Dynamic foreground

of foreground detection ($\text{F-measure} > 0.95$) over 200 trials as a 2D function of σ_F^2 and W. The result of PCP is also shown for comparison. As we can see, DECOLOR can achieve accurate detection with a high probability over a wide range of conditions, except for the upper left corner where W is large and σ_F^2 is small, which represents the case of large and textureless foreground. In practice, the interior motion of a textureless object is undetectable. Thus, its interior region will remain unchanged for a relatively long time if the object is large or moving slowly. In this case, the interior part of the foreground may fit into the low-rank model, which makes DECOLOR fail.

5.2 Real Sequences

We test DECOLOR on real sequences from public datasets for background subtraction, motion segmentation and dynamic texture detection. Please refer to Table 1 for the details of each sequence.

5.2.1 *Comparison to Sparse Signal Recovery*

As discussed in Section 3.4.2, a key difference between DECOLOR and sparse signal recovery is the assumption on availability of training sequences. Background subtraction via sparse signal recovery requires a set of background images without foreground, which is not always available, especially for surveillance of crowded scenes. Fig. 5a gives such a sequence, clipped from the start of an indoor surveillance video, where the couple is always in the scene.

Fig. 5b shows the results of the 24th frame. For sparse signal recovery, we apply the ProxFlow algorithm[2] [41] to solve the model in (18). The previous 23 frames are used as the bases (Φ in (18)). Since the subspace spanned by previous frames also includes foreground objects, ProxFlow cannot recover the background and gives inaccurate segmentation. Instead, DECOLOR can estimate a clean background from occluded data. In practice, DECOLOR can be used for background initialization. For example, the last column in Fig. 5b shows the results of running ProxFlow with Φ being low-rank $\hat{B}$ learned by DECOLOR. That is, we use the background images recovered by DECOLOR as the training images for background subtraction. We can see that the results are improved apparently.

5.2.2 *Background Estimation*

In this part, we test DECOLOR on several real sequences selected from public datasets of background subtraction.

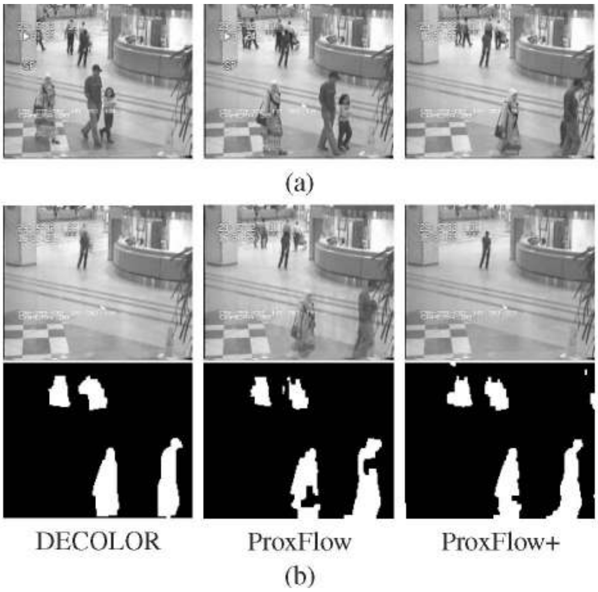


Fig. 5. An example illustrating the difference between DECOLOR and sparse signal recovery. (a) The first, middle, and last frames of a sequence of 24 images. Several people are walking and are continuously presented in the scene. (b) The estimated background (top) and segmentation (bottom) corresponding to the last frame. ProxFlow means sparse signal recovery by solving (18) with the ProxFlow algorithm [41], where the first 23 frames are used as the basis matrix Φ in (18). ProxFlow+ means applying ProxFlow with bases Φ being the low-rank matrix $\hat{B}$ learned by DECOLOR.

Since we aim to evaluate the ability of algorithms in detecting moving objects at the start of videos, we focus on short clips composed of beginning frames of videos. All examples in Fig. 6 have only 24 or 48 frames, corresponding to 1 or 2 seconds for a frame rate of 24 fps. We compare DECOLOR with three methods that are simple in implementation but effective in practice. The first one is PCP [13], which is the state-of-the-art algorithm for RPCA. The second method is median filtration, a baseline method for unimodal background modeling. The median intensity value around each pixel is computed, forming a background image. Then, each frame is subtracted by the background image and the difference is thresholded to generate a foreground mask. The advantage of using median rather than mean is that it is a more robust estimator to avoid blending pixel values, which is more proper for background estimation [11]. The third method is mixture of Gaussians [28]. It is popularly used for multimodal background modeling and has proven to be very competitive compared with other more sophisticated techniques for background subtraction [7], [65].

The sequences and results are presented in Fig. 6. The first example shows an office with two people walking around. Although the objects are large and always presented in all frames, DECOLOR recovers the background and outputs a foreground mask accurately. Notice that the results are direct outputs of Algorithm 1 without any postprocessing. The results of PCP are relatively unsatisfactory. Ghosts of the foreground remain in the recovered background. This is because the ℓ_1-penalty used in PCP is not robust enough to remove the influence of contiguous occlusion. Such corruption of extracted background will result in false detections as shown in the segmentation result. Moreover, without the

[2] The code is available at http://www.di.ens.fr/willow/SPAMS/.

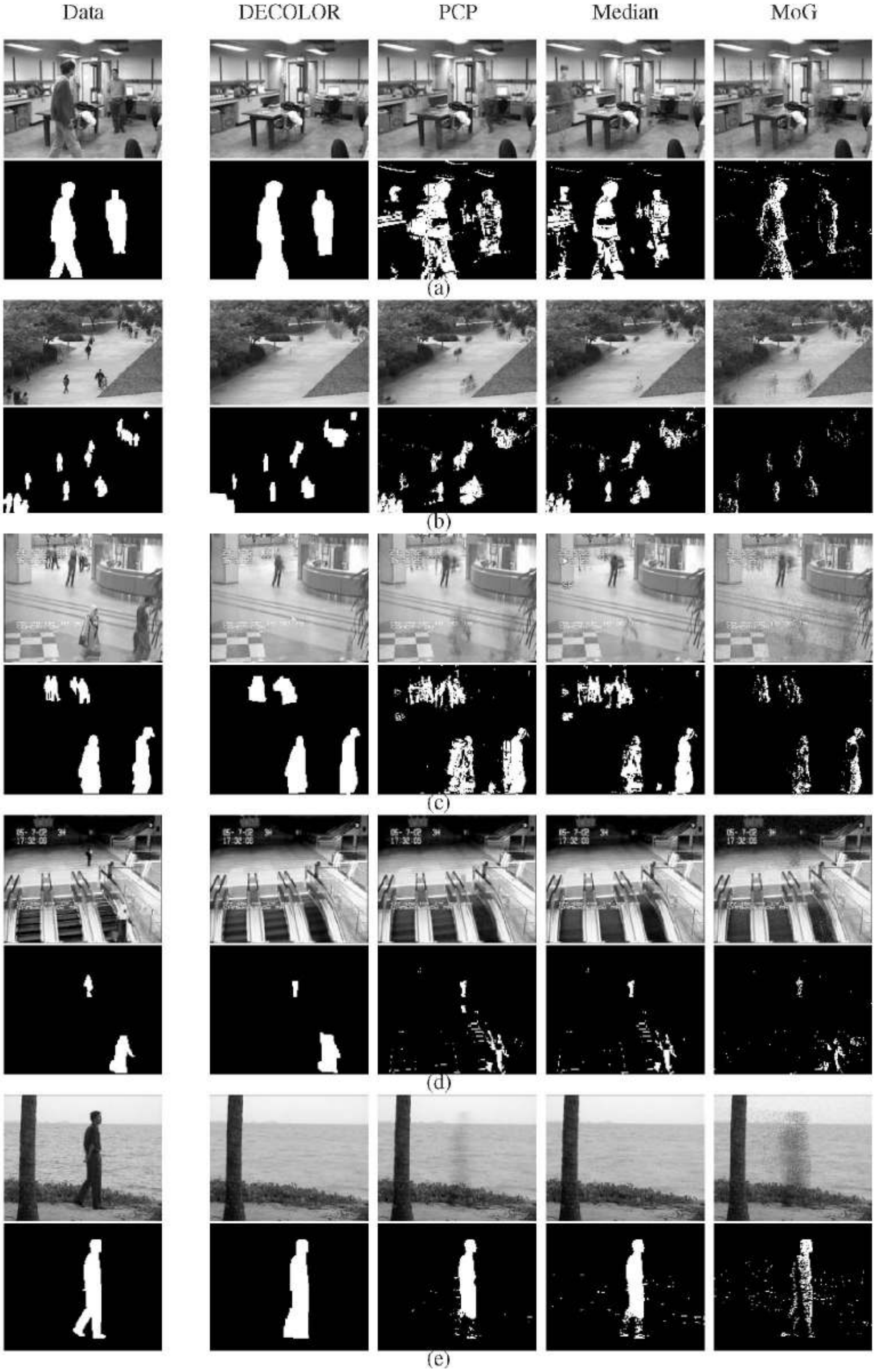


Fig. 6. Five subsequences of surveillance videos. Sequence information is given in Table 1. The last frame of each sequence and its manual segmentation are shown in Column 1. The corresponding results by four methods are presented from Columns 2 to 5, respectively. The top panel is the estimated background and the bottom panel is the segmentation.

smoothness constraint, occasional light changes (e.g., near the boundary of fluorescent lamps) or video noises give rise to small pieces of falsely detected regions. The results of median filtration depend on how long each pixel is taken by foreground. Thus, from the recovered background of median filtration, we can find that the man near the door is clearly removed while the man turning at the corner leaves a ghost. Despite scattered artifacts, MoG gives fewer false positives due to its multimodal modeling of background. However, blending of foreground intensity can be seen obviously in the recovered background, which results in more false negatives in the foreground mask, e.g., the

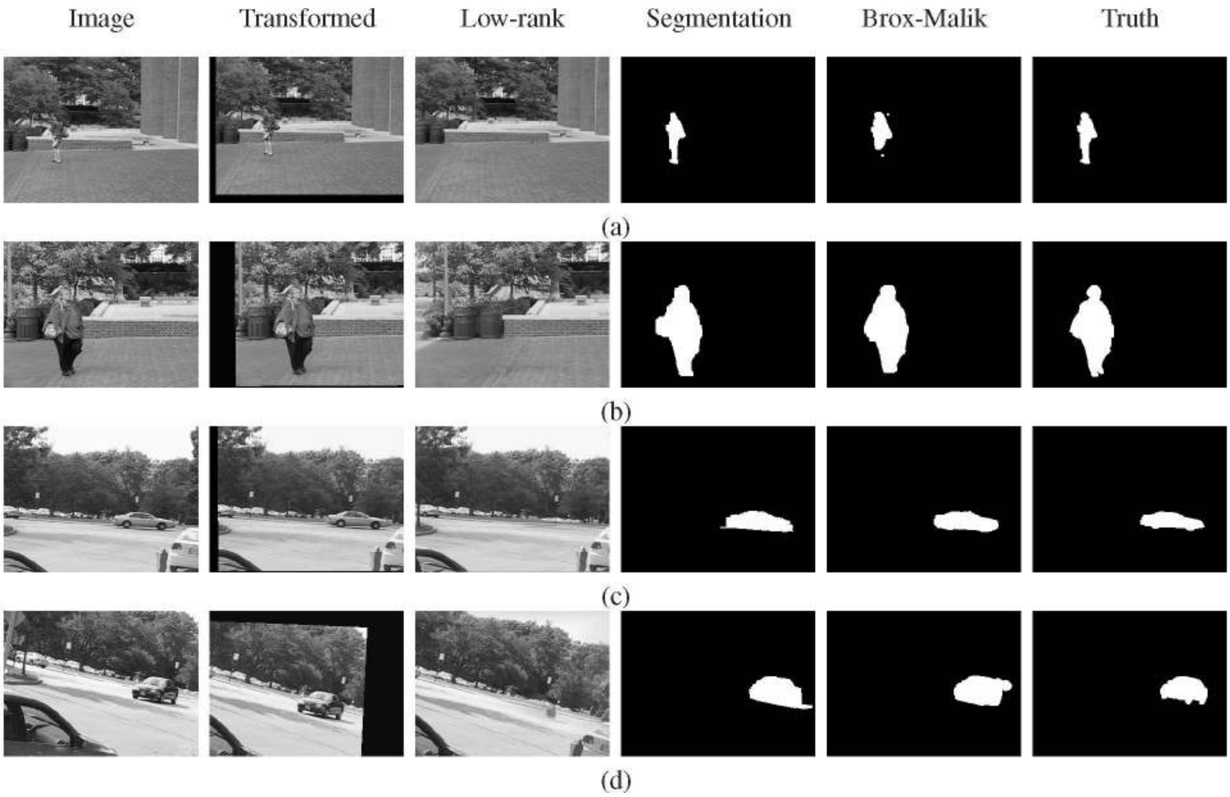


Fig. 7. Four sequences captured by moving cameras. Sequence information is given in Table 1. Only the last frame of each sequence and the corresponding results are shown. Columns 2-4 present the results of DECOLOR, i.e., the transformed image, the estimated background, and the foreground mask. Column 5 shows the results given by Brox and Malik's algorithm for motion segmentation [24]. The last column shows the ground truth.

interior region of objects. Similar results can be found in the next two examples.

The last two examples include dynamic background. Fig. 6d presents a sequence clipped from a surveillance video of an airport, which is very challenging because the background involves a running escalator. Although the escalator is moving, it is recognized as a part of background by DECOLOR since its periodical motion gives repeated patterns. As we can see, the structure of the escalator is maintained in the background recovered by DECOLOR or PCP. This demonstrates the ability of low-rank representation to model dynamic background. Fig. 6e gives another example with a water surface as background. Similarly, the low-rank modeling of background gives better results with fewer false detections on the water surface and DECOLOR obtains a cleaner background compared against PCP.

We also give a quantitative evaluation for the segmentation results shown in Fig. 6. The manual annotation is used as ground truth and the F-measure is calculated. As shown in Table 2, DECOLOR outperforms other approaches on all sequences.

5.2.3 Moving Cameras

Next, we demonstrate the potential of DECOLOR applied to motion segmentation problems using the Berkeley motion segmentation dataset.[3] We use two *people* sequences and 12 *car* sequences, which are specialized for short-term analysis. Each sequence has several annotated frames as the ground truth for segmentation. Fig. 7 shows several examples and the results of DECOLOR. The transformed images $D \circ \hat{\tau}$ are shown in Column 2. Notice the extrapolated regions shown in black near the borders of these images. To minimize the influence of this numerical error, we constrain these pixels to be background when estimating S, but consider them as missing entries when estimating B. Fig. 7 demonstrates that DECOLOR can align the images, learn a background model, and detect objects correctly.

For comparison, we also test the motion segmentation algorithm recently developed by Brox and Malik [24]. The Brox-Malik algorithm analyzes the point trajectories along the sequence and segment them into clusters. To obtain pixel-level segmentation, the variational method [26] can be applied to turn the trajectory clusters into dense regions. This additional step makes use of the color and edge

TABLE 2
Quantitative Evaluation (F-Measure) on the Sequences Shown in Fig. 6

Sequence	DECOLOR	PCP	Median	MoG
Fig. 6(a)	0.93	0.62	0.67	0.50
Fig. 6(b)	0.82	0.66	0.71	0.35
Fig. 6(c)	0.92	0.70	0.79	0.50
Fig. 6(d)	0.82	0.49	0.51	0.36
Fig. 6(e)	0.91	0.83	0.86	0.47

[3] http://lmb.informatik.uni-freiburg.de/resources/datasets/moseg.en.html.

TABLE 3
Quantitative Evaluation Using the Sequences from the Berkeley Motion Segmentation Dataset [24]

	DECOLOR		Brox-Malik [24]	
Sequence	Precision	Recall	Precision	Recall
Fig. 7(a)	93.6%	93.3%	89.0%	77.5%
Fig. 7(b)	92.5%	96.5%	91.7%	89.2%
Fig. 7(c)	83.7%	98.4%	82.4%	99.4%
Fig. 7(d)	72.0%	98.0%	76.4%	99.8%
Overall	81.8%	90.8%	80.8%	99.2%

The overall result is the median value over all people and car sequences.

information in images [26], while DECOLOR only uses the motion cue and directly generates the segmentation.

Quantitatively, we calculate the precision and recall of foreground detection, as shown in Table 3. In summary, for most sequences with moderate camera motion, the performance of DECOLOR is competitive. On the *people* sequences, DECOLOR performs better. The feet of the lady are not detected by the Brox-Malik algorithm. The reason is that the Brox-Malik algorithm relies on correct motion tracking and clustering [26], which is difficult when the object is small and moving nonrigidly. Instead, DECOLOR avoids the complicated motion analysis. However, DECOLOR works poorly on the cases where the background is a 3D scene with a large depth and the camera moves a lot, e.g., the sequences named *cars9* and *cars10*. This is because the parametric motion model used in DECOLOR can only compensate for the planar background motion.

5.2.4 Dynamic Foreground

Dynamic texture segmentation has drawn some attention in recent computer vision research [18], [20]. While we have shown that DECOLOR can model periodically varying textures like escalators or water surfaces as background, it is also able to detect fast changing textures whose motion has little periodicity and cannot be modeled as low rank. Fig. 8 shows such an example, where the smoke is detected as foreground. Here, the background behind the smoke cannot be recovered since it is always occluded.

5.2.5 Computational Cost

Our algorithm is implemented in Matlab. All experiments are run on a desktop PC with a 3.4 GHz Intel i7 CPU and 3 GB RAM. Since the graph cut is operated for each frame separately, as discussed in Section 3.3.2, the dominant cost comes from the computation of SVD in each iteration. The cpu times of DECOLOR for sequences in Fig. 6 are 26.2, 13.3, 14.1, 11.4, and 14.4 seconds, while those of PCP are 26.8, 38.0, 15.7, 39.1, and 21.9 seconds, respectively. All results are obtained with a convergence precision of 10^{-4}. The memory costs of DECOLOR and PCP are almost the same since both of them need to compute SVD. The peak values of memory used in DECOLOR for sequences in Figs. 6a and 7b are around 65 MB and 210 MB, respectively.

6 Discussion

In this paper, we propose a novel framework named DECOLOR to segment moving objects from image sequences. It avoids complicated motion computation by formulating the problem as outlier detection and makes use of the low-rank modeling to deal with complex background.

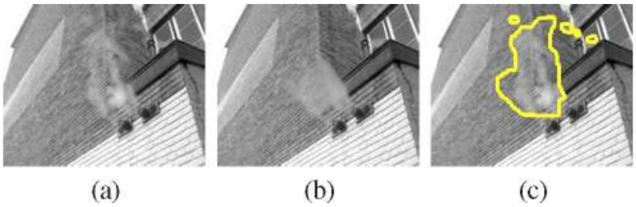


Fig. 8. An example of smoke detection. (a) Sample frame. (b) Estimated background. (c) Segmentation.

We established the link between DECOLOR and PCP. Compared with PCP, DECOLOR uses the nonconvex penalty and MRFs for outlier detection, which is more greedy to detect outlier regions that are relatively dense and contiguous. Despite its satisfactory performance in our experiments, DECOLOR also has some disadvantages. Since DECOLOR minimizes a nonconvex energy via alternating optimization, it converges to a local optimum with results depending on initialization of $\hat{S}$, while PCP always minimizes its energy globally. In all our experiments, we simply start from $\hat{S} = \mathbf{0}$. Also, we have tested other random initialization of $\hat{S}$ and it generally converges to a satisfactory result. This is because the SOFT-IMPUTE step will output similar results for each randomly generated $\hat{S}$ as long as $\hat{S}$ is not too dense.

As illustrated in Section 5.1.3, DECOLOR may misclassify unmoved objects or large textureless regions as background since they are prone to entering the low-rank model. To address these problems, incorporating additional models such as object appearance or shape prior to improve the power of DECOLOR can be further explored in future.

Currently, DECOLOR works in a batch mode. Thus, it is not suitable for real-time object detection. In the future, we plan to develop the online version of DECOLOR that can work incrementally, e.g., the low-rank model extracted from beginning frames may be updated online when new frames arrive.

Acknowledgments

This research was supported by the grant ITS/180/11 from the Innovation and Technology Commission of the Hong Kong SAR government.

References

[1] A. Yilmaz, O. Javed, and M. Shah, “Object Tracking: A Survey,” *ACM Computing Surveys*, vol. 38, no. 4, pp. 1-45, 2006.

[2] T. Moeslund, A. Hilton, and V. Kruger, “A Survey of Advances in Vision-Based Human Motion Capture and Analysis,” *Computer Vision and Image Understanding*, vol. 104, nos. 2/3, pp. 90-126, 2006.

[3] C. Papageorgiou, M. Oren, and T. Poggio, “A General Framework for Object Detection,” *Proc. IEEE Int'l Conf. Computer Vision*, p. 555, 1998.

[4] P. Viola, M. Jones, and D. Snow, “Detecting Pedestrians Using Patterns of Motion and Appearance,” *Int'l J. Computer Vision*, vol. 63, no. 2, pp. 153-161, 2005.

[5] H. Grabner and H. Bischof, "On-Line Boosting and Vision," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* pp. 260-267, 2006.

[6] B. Babenko, M.-H. Yang, and S. Belongie, "Robust Object Tracking with Online Multiple Instance Learning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 8, pp. 1619-1632, Aug. 2011.

[7] M. Piccardi, "Background Subtraction Techniques: A Review," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics,* 2004.

[8] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and Practice of Background Maintenance," *Proc. IEEE Int'l Conf. Computer Vision,* 1999.

[9] R. Vidal and Y. Ma, "A Unified Algebraic Approach to 2-D and 3-D Motion Segmentation," *Proc. European Conf. Computer Vision,* 2004.

[10] D. Cremers and S. Soatto, "Motion Competition: A Variational Approach to Piecewise Parametric Motion Segmentation," *Int'l J. Computer Vision,* vol. 62, no. 3, pp. 249-265, 2005.

[11] D. Gutchess, M. Trajkovics, E. Cohen-Solal, D. Lyons, and A. Jain, "A Background Model Initialization Algorithm for Video Surveillance," *Proc. IEEE Int'l Conf. Computer Vision,* 2001.

[12] V. Nair and J. Clark, "An Unsupervised, Online Learning Framework for Moving Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 317-324, 2004.

[13] E. Candes, X. Li, Y. Ma, and J. Wright, "Robust Principal Component Analysis?" *J. ACM,* vol. 58, article 11, 2011.

[14] S. Li, *Markov Random Field Modeling in Image Analysis.* Springer-Verlag, 2009.

[15] M. Black and P. Anandan, "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields," *Computer Vision and Image Understanding,* vol. 63, no. 1, pp. 75-104, 1996.

[16] T. Amiaz and N. Kiryati, "Piecewise-Smooth Dense Optical Flow via Level Sets," *Int'l J. Computer Vision,* vol. 68, no. 2, pp. 111-124, 2006.

[17] T. Brox, A. Bruhn, and J. Weickert, "Variational Motion Segmentation with Level Sets," *Proc. European Conf. Computer Vision,* 2006.

[18] A. Chan and N. Vasconcelos, "Layered Dynamic Textures," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 10, pp. 1862-1879, Oct. 2009.

[19] G. Doretto, D. Cremers, P. Favaro, and S. Soatto, "Dynamic Texture Segmentation," *Proc. IEEE Int'l Conf. Computer Vision,* 2003.

[20] S. Fazekas, T. Amiaz, D. Chetverikov, and N. Kiryati, "Dynamic Texture Detection Based on Motion Analysis," *Int'l J. Computer Vision,* vol. 82, no. 1, pp. 48-63, 2009.

[21] S. Beauchemin and J. Barron, "The Computation of Optical Flow," *ACM Computing Surveys,* vol. 27, no. 3, pp. 433-466, 1995.

[22] R. Tron and R. Vidal, "A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2007.

[23] Y. Sheikh, O. Javed, and T. Kanade, "Background Subtraction for Freely Moving Cameras," *Proc. IEEE Int'l Conf. Computer Vision,* 2009.

[24] T. Brox and J. Malik, "Object Segmentation by Long Term Analysis of Point Trajectories," *Proc. European Conf. Computer Vision,* 2010.

[25] R. Vidal, "Subspace Clustering," *IEEE Signal Processing Magazine,* vol. 28, no. 2, pp. 52-68, Mar. 2011.

[26] P. Ochs and T. Brox, "Object Segmentation in Video: A Hierarchical Variational Approach for Turning Point Trajectories Into Dense Regions," *Proc. IEEE Int'l Conf. Computer Vision,* 2011.

[27] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 780-785, July 1997.

[28] C. Stauffer and W. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1999.

[29] A.M. Elgammal, D. Harwood, and L.S. Davis, "Non-Parametric Model for Background Subtraction," *Proc. European Conf. Computer Vision,* 2000.

[30] A. Mittal and N. Paragios, "Motion-Based Background Subtraction Using Adaptive Kernel Density Estimation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2004.

[31] T. Matsuyama, T. Ohya, and H. Habe, "Background Subtraction for Non-Stationary Scenes," *Proc. Asian Conf. Computer Vision,* 2000.

[32] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-Time Foreground-Background Segmentation Using Codebook Model," *Real-Time Imaging,* vol. 11, no. 3, pp. 172-185, 2005.

[33] N. Friedman and S. Russell, "Image Segmentation in Video Sequences: A Probabilistic Approach," *Proc. Conf. Uncertainty in Artificial Intelligence,* 1997.

[34] J. Rittscher, J. Kato, S. Joga, and A. Blake, "A Probabilistic Background Model for Tracking," *Proc. European Conf. Computer Vision,* 2000.

[35] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background Modeling and Subtraction of Dynamic Scenes," *Proc. IEEE Int'l Conf. Computer Vision,* 2003.

[36] J. Zhong and S. Sclaroff, "Segmenting Foreground Objects from a Dynamic Textured Background via a Robust Kalman Filter," *Proc. IEEE Int'l Conf. Computer Vision,* 2003.

[37] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, "Sparse Representation for Computer Vision and Pattern Recognition," *Proc. IEEE,* vol. 98, no. 6, pp. 1031-1044, June 2010.

[38] N. Oliver, B. Rosario, and A. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 831-843, Aug. 2000.

[39] V. Cevher, M. Duarte, C. Hegde, and R. Baraniuk, "Sparse Signal Recovery Using Markov Random Fields," *Proc. Advances in Neural Information and Processing Systems,* 2008.

[40] J. Huang, X. Huang, and D. Metaxas, "Learning with Dynamic Group Sparsity," *Proc. IEEE Int'l Conf. Computer Vision,* 2009.

[41] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach, "Network Flow Algorithms for Structured Sparsity," *Proc. Advances in Neural Information and Processing Systems,* 2010.

[42] H. Wang and D. Suter, "A Novel Robust Statistical Method for Background Initialization and Visual Surveillance," *Proc. Asian Conf. Computer Vision,* 2006.

[43] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 6, pp. 721-741, Nov. 1984.

[44] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization," *SIAM Rev.,* vol. 52, no. 3, pp. 471-501, 2010.

[45] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral Regularization Algorithms for Learning Large Incomplete Matrices," *J. Machine Learning Research,* vol. 11, pp. 2287-2322, 2010.

[46] J. Cai, E. Candès, and Z. Shen, "A Singular Value Thresholding Algorithm for Matrix Completion," *SIAM J. Optimization,* vol. 20, pp. 1956-1982, 2010.

[47] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 11, pp. 1222-1239, Nov. 2001.

[48] V. Kolmogorov and R. Zabih, "What Energy Functions Can Be Minimized via Graph Cuts?" *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 2, pp. 147-159, Feb. 2004.

[49] F. De La Torre and M. Black, "A Framework for Robust Subspace Learning," *Int'l J. Computer Vision,* vol. 54, no. 1, pp. 117-142, 2003.

[50] Q. Ke and T. Kanade, "Robust l1 Norm Factorization in the Presence of Outliers and Missing Data by Alternative Convex Programming," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2005.

[51] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma, "Stable Principal Component Pursuit," *Proc. Int'l Symp. Information Theory,* 2010.

[52] Y. She and A.B. Owen, "Outlier Detection Using Nonconvex Penalized Regression," *J. Am. Statistical Assoc.,* vol. 106, pp. 626-639, 2010.

[53] P. Zhao and B. Yu, "On Model Selection Consistency of Lasso," *J. Machine Learning Research,* vol. 7, pp. 2541-2563, 2006.

[54] R. Mazumder, J. Friedman, and T. Hastie, "Sparsenet: Coordinate Descent with Non-Convex Penalties," *J. Am. Statistical Assoc.,* 2011.

[55] D. Donoho, "Compressed Sensing," *IEEE Trans. Information Theory,* vol. 52, no. 4, pp. 1289-1306, Apr. 2006.

[56] Z. Zhou, A. Wagner, H. Mobahi, J. Wright, and Y. Ma, "Face Recognition with Contiguous Occlusion Using Markov Random Fields," *Proc. IEEE Int'l Conf. Computer Vision,* 2010.

[57] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust Alignment by Sparse and Low-Rank Decomposition for Linearly Correlated Images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[58] M. Yuan and Y. Lin, "Model Selection and Estimation in Regression with Grouped Variables," *J. Royal Statistical Soc.,* vol. 68, no. 1, pp. 49-67, 2006.

[59] P. Zhao, G. Rocha, and B. Yu, "The Composite Absolute Penalties Family for Grouped and Hierarchical Variable Selection," *The Annals of Statistics,* vol. 37, no. 6A, pp. 3468-3497, 2009.

[60] R. Szeliski, *Computer Vision: Algorithms and Applications.* Springer, 2010.

[61] J. Odobez and P. Bouthemy, "Robust Multiresolution Estimation of Parametric Motion Models," *J. Visual Comm. Image Representation,* vol. 6, no. 4, pp. 348-365, 1995.

[62] J. Davis and M. Goadrich, "The Relationship between Precision-Recall and Roc Curves," *Proc. Int'l Conf. Machine Learning,* 2006.

[63] E. Candes, M. Wakin, and S. Boyd, "Enhancing Sparsity by Reweighted L1 Minimization," *J. Fourier Analysis Applications,* vol. 14, no. 5, pp. 877-905, 2008.

[64] L. Li, W. Huang, I. Gu, and Q. Tian, "Statistical Modeling of Complex Backgrounds for Foreground Object Detection," *IEEE Trans. Image Processing,* vol. 13, no. 11, pp. 1459-1472, Nov. 2004.

[65] D. Parks and S. Fels, "Evaluation of Background Subtraction Algorithms with Post-Processing," *Proc. IEEE Int'l Conf. Advanced Video and Signal Based Surveillance,* pp. 192-199, 2008.

Xiaowei Zhou received the bachelor's degree in optical engineering from Zhejiang University, China, in 2008. He is currently working toward the PhD degree in the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology. His research interests include computer vision and medical image analysis. He is a student member of the IEEE and the IEEE Computer Society.

Can Yang received the bachelor's and master's degrees in automatic control from Zhejiang University, China, in 2003 and 2006, respectively. He received the PhD degree in electronic and computer engineering from the Hong Kong University of Science and Technology in 2011. Now, he is working as a postdoctoral researcher at Yale University, New Haven, Connecticut. His research interests include bioinformatics, machine learning, and pattern recognition.

Weichuan Yu received the PhD degree in computer vision and image analysis from the University of Kiel, Germany, in 2001. He is currently an associate professor in the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology. He is interested in computational analysis problems with biological and medical applications. He has published papers on a variety of topics, including bioinformatics, computational biology,biomedical imaging, signal processing, pattern recognition, and computer vision. He is a member of the IEEE.