Efficient Background Modeling Based on Sparse Representation and Outlier Iterative Removal

Linhao Li, Ping Wang, Qinghua Hu, *Senior Member, IEEE*, and Sijia Cai

Abstract—**Background modeling is a critical component for various vision-based applications. Most traditional methods tend to be inefficient when solving large-scale problems. In this paper, we introduce sparse representation into the task of large-scale stable-background modeling, and reduce the video size by exploring its discriminative frames. A cyclic iteration process is then proposed to extract the background from the discriminative frame set. The two parts combine to form our sparse outlier iterative removal (SOIR) algorithm. The algorithm operates in tensor space to obey the natural data structure of videos. Experimental results show that a few discriminative frames determine the performance of the background extraction. Furthermore, SOIR can achieve high accuracy and high speed simultaneously when dealing with real video sequences. Thus, SOIR has an advantage in solving large-scale tasks.**

Index Terms—**Alternating direction multipliers (ADMs) method, background modeling, Markov random field (MRF), principal component pursuit (PCP), sparse representation, tensor analysis.**

I. Introduction

The background modeling of a video sequence is a key part in many vision-based applications, such as real-time tracking [1], [2], information retrieval, and video surveillance [3], [4]. In a video sequence, some scenes will remain nearly constant, even though they may be polluted by noise [5]. The invariable aspect is the background. A model for extracting the background is an important tool that can help us handle a video sequence, especially one taken in a public area [6]. Background modeling is also an essential step in many foreground detection tasks [7]–[9]. Once the background is extracted, we can detect or even track the foreground information by comparing a new frame with the learned background model [4].

There are two challenges to a background modeling algorithm. First, although we consider the background to be stationary, it is often interfered with by certain factors, such as fluttering flags, waving leaves, or rippling water [10]. In addition, other issues, such as signal noise, sudden lighting variations, and shadows [11], [12], may prevent us from distinguishing the background from a video sequence. Second, the data on practical problems are increasing with the development of new technologies and improvements in the equipment used. However, there is also an increasing demand for efficient background modeling techniques, and fast tracking of massive video sequences is required for certain practical tasks like crime detection and recognition. As a result, it has become an urgent task to develop an efficient and robust algorithm for practical background modeling.

A large number of background modeling methods have been reported in the literature over the past decades. Most researchers have regarded a series of pixel values as features and set up pixel-wise models. Initially, each pixel-value series is modeled using a Gaussian distribution, e.g., the single Gaussian (SG) model developed in [13] and the multiple of Gaussian (MOG) model developed in [14]. Some improved Gaussian-based algorithms [15]–[17] also achieved a high level of performance in the few years following the release of the above models. In addition, clustering methods have also been used to model a background, e.g., codebook [18], [19] and time-series clustering [20]. Furthermore, a nonparametric method was proposed in [21] and improved in [22], and has shown a competitive performance. The Visual Background Extractor (ViBe) was recently proposed in 2012 and later improved, and performs better than most popular pixel-wise techniques [8], [23]. These methods solve the background problem by building a model for each pixel and initializing the models during the training process. High accuracy is obtained if sufficient training data are provided, but more training data means additional training time.

Another type of modeling technique is to set up the model at region level. Some works have focused on the local region, and different local features have been proposed [11], [24]–[26]. In addition, global-region-based algorithms have also been proposed. Oliver *et al.* [27] first modeled a background, using a principal component analysis (PCA), i.e., they modeled the background by projecting high-dimensional data into a lower dimensional subspace. Robust PCA developed in [28] and principal component pursuit (PCP) developed in [29] have shown their superiority over the original PCA. Based on these models, heuristic background

Manuscript received February 17, 2014; revised May 2, 2014, June 17, 2014, August 22, 2014, and September 30, 2014; accepted November 26, 2014. Date of publication December 12, 2014; date of current version February 3, 2016. This work was supported in part by the National Natural Science Foundation of China under Grants 51275348, 61379014, 6122210, and 61432011, and in part by the New Century Excellent Talents in University under Grant NCET-12-0399. This paper was recommended by Associate Editor Y. Fu.

L. Li is with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: lilinhao@tju.edn.cn).

P. Wang is with the School of Science, Tianjin University, Tianjin 300072, China, and also with Center for Applied Mathematics, Tianjin University, Tianjin 300072, China, (e-mail: pwangtjun@gmail.com).

Q. Hu is with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China, and also with the Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, Tianjin 300072, China (e-mail: huqinghua@tju.edu.cn).

S. Cai is with the School of Science, Tianjin University, Tianjin 300072, China (e-mail: caisj@tju.edu.cn).

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

Digital Object Identifier 10.1109/TCSVT.2014.2380195

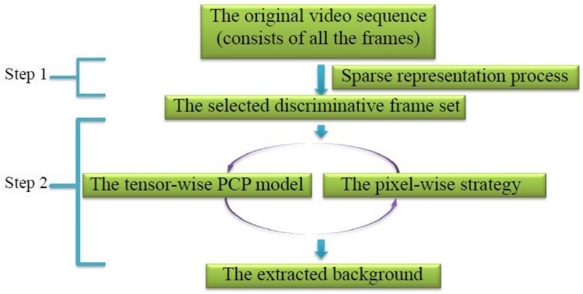

The original video sequence (consists of all the frames)
Step 1
Sparse representation process
The selected discriminative frame set
Step 2
The tensor-wise PCP model
The pixel-wise strategy
The extracted background

Fig. 1. Framework of the SOIR algorithm.

methods have also been introduced [30], [31], [33]. These PCA-based models omit the training process and use data to extract the background directly. However, singular value decomposition (SVD) is an inevitable time-consuming step in a PCA-based model, and thus, these models are limited in large-scale tasks because their speed and memory requirements are all sensitive to the scale of the data.

Sparse representation and dictionary learning is also an important region-based method. It is widely employed in the tasks of computer vision, such as face recognition [32], [34], [35], classification [34], [36], and denoising [37]. Some researchers have introduced sparse representation into background modeling [38], [39]. They modeled the background by the dictionary and regarded the foreground as noise. In addition, they made some assumptions of independence among different pixels. However, these assumptions fail in many practical tasks where the foreground region is usually not sparse and some pixels are highly correlated.

In this paper, first, we use sparse representation to reduce the video size by exploring the discriminative frames of the video, instead of modeling the background directly. No assumption is needed in this process. We then extract the background from these discriminative frames using a PCP-based cyclic iteration. These two steps combine to form our algorithm, i.e., sparse outlier iterative removal (SOIR) algorithm. The framework of the algorithm is shown in Fig. 1.

SOIR meets the demand of global-region-based background models on solving large-scale problems. For our algorithm, we rebuild the PCP model based on a rank-1 hypothesis. Moreover, our algorithm operates in a tensor space to obey the natural data structure of the videos. We detect foreground objects using the Markov random field (MRF) once the background is extracted. Experimental results show that our algorithm can achieve high accuracy and high speed simultaneously when dealing with real-life video sequences.

The main contributions of this paper are summarized as follows.

1) We utilize sparse representation to reduce the size of the video by exploring its discriminative frames. Instead of using all frames to model the background, we simply use the discriminative frames. In this way, our model can meet the demands of many practical background modeling problems in terms of speed and memory.
2) The cyclic iteration process is composed of a tensor-wise model and a pixel-wise strategy. In a general case, a tensor-wise process always considers the overall information, whereas a pixel-wise process pays more attention to particular information. Our algorithm achieves high accuracy by taking full advantage of both processes.
3) The tensor-wise model in the cyclic iteration is a PCP model and is robust to general noises [29]. Differing from previous works, the vectorized static background in our algorithm is explicit rank-1, instead of just being low rank. To constrain this, we propose a new space $\mathcal{R}^{(4)}$, where the background actually lies. Owing to the rank-1 hypothesis, SVD is nonessential.

The remainder of this paper is organized as follows. Section II introduces some preliminary works. Section III provides the formulation and convergence of the SOIR algorithm. Section IV presents the foreground detection method. Section V describes the experimental results. Finally, Section VI provides some concluding remarks regarding our research.

II. Preliminary Work

The PCP model and tensor theory play key roles in our algorithm. Here, we introduce some basic works of both.

A. Principal Component Pursuit

Low-rank matrix recovery is the key problem in many practical tasks, including background modeling. A given data matrix M is the superposition of a low-rank matrix L and a sparse matrix S, i.e., $M = L + S$. PCA is an effective way to solve this problem, but the brittleness of the original PCA model with respect to grossly corrupted observations jeopardizes its validity [29].

Candès *et al.* [29] recently proved that one can recover matrix L and the sparse matrix S precisely under mild conditions. This model, known as PCP, can be formulated as

$$\begin{aligned}&\min_{L,S} \ \|L\|_* + \lambda\|S\|_1\\ &\text{s.t. } L + S = M\end{aligned} \tag{1}$$

where λ is a regularization parameter and $\|\cdot\|_*$ and $\|\cdot\|_1$ denote the nuclear norm (sum of singular values) and the l_1-norm (sum of the absolute values of the matrix elements), respectively.

Model (1) was modified to solve the background modeling problem [31], [33]. All of the original works modeled the background using a low-rank matrix. In contrast, we consider the background as a rank-1 matrix.

B. Tensors Theory

A tensor is a multidimensional array. More formally, an N-way or an N-order tensor is an element of the tensor product of N vector spaces, each of which has its own coordinate system [40], [41]. Intuitively, a vector is a first-order tensor, while a matrix is a second-order tensor. In this paper, in addition to the specific instructions, we denote vectors by lowercase letters, e.g., r, and matrices by uppercase letters, e.g., R. In addition, a higher order tensor is denoted in boldface type, e.g., $\mathbf{R}$. The space of all tensors is denoted by a swash font, e.g., $\mathcal{R}$. We denote the space of all N-order tensors by $\mathcal{R}_N$, $\mathcal{R}_N = \mathbb{R}^{K_1\times\cdots\times K_N}$, $N = 1, 2, 3, 4, \ldots$.

A tensor can be multiplied by a matrix, which is also known as the n-mode (matrix) product [41]. The n-mode product of tensor $\mathbf{X} \in \mathbb{R}^{K_1\times\cdots\times K_N}$ with matrix $U \in \mathbb{R}^{J\times K_i}$ is denoted by $\mathbf{X}\times_i U$ and is of size $K_1\times\cdots\times K_{i-1}\times J\times K_{i+1}\times\cdots\times K_N$. Element wise, we have

$$(\mathbf{X}\times_i U)_{k_1,\ldots,k_{i-1}jk_{i+1},\ldots,k_N} = \sum_{k_i=1}^{K_i} \mathbf{X}_{k_1,\ldots,k_N} U_{jk_i} \tag{2}$$

where $k_i \in [1,\ldots,K_i]$ $(i = 1,\ldots,N)$; $j \in [1,\ldots,J]$.

III. Sparse Outlier Iterative Removal Algorithm

In this section, we focus on modeling the background of a video sequence. We use $\mathbf{D}$ to denote a video and assume that there are N frames in $\mathbf{D}$. Each colorful frame is a third-order tensor by nature, and the jth frame is denoted by $\mathbf{I}_j \in \mathbb{R}^{m\times n\times 3}$. Then, $\mathbf{D} = [\mathbf{I}_1,\ldots,\mathbf{I}_N] \in \mathbb{R}^{m\times n\times 3\times N}$. In addition, we use $\mathbf{B} = [\mathbf{B}_{\mathbf{I}_1},\ldots,\mathbf{B}_{\mathbf{I}_N}]$ and $\mathbf{A} = [\mathbf{A}_{\mathbf{I}_1},\ldots,\mathbf{A}_{\mathbf{I}_N}]$ to denote the background and foreground of video $\mathbf{D}$, respectively.

We first analyze the components of a video. In the video, the background is covered by the foreground objects. We denote the foreground region by $\mathbf{\Omega}$, and the outside region by $\overline{\mathbf{\Omega}}$. Let $\mathcal{P}_\Omega$ be an orthogonal projector onto the span of the tensors vanishing outside of $\mathbf{\Omega}$. Then, for an arbitrary frame $\mathbf{I}_j$, the (x, y, z)th component of $\mathcal{P}_\Omega(\mathbf{I}_j)$ is equal to $(\mathbf{I}_j)_{xyz}$ if $(x, y, z) \in \mathbf{\Omega}$, and is zero, otherwise. Thus, the video can be expressed as

$$\mathbf{D} = \mathcal{P}_{\overline{\Omega}}(\mathbf{B}) + \mathcal{P}_\Omega(\mathbf{A}) \tag{3}$$

where $\mathcal{P}_{\overline{\Omega}}(\mathbf{B}) = [\mathcal{P}_{\overline{\Omega}}(\mathbf{B}_{\mathbf{I}_1}),\ldots,\mathcal{P}_{\overline{\Omega}}(\mathbf{B}_{\mathbf{I}_N})]$ and $\mathcal{P}_\Omega(\mathbf{A}) = [\mathcal{P}_\Omega(\mathbf{A}_{\mathbf{I}_1}),\ldots,\mathcal{P}_\Omega(\mathbf{A}_{\mathbf{I}_N})]$. Actually, $\mathcal{P}_\Omega(\mathbf{A}_{\mathbf{I}_i}) = \mathbf{A}_{\mathbf{I}_i}$ because $\mathbf{\Omega}$ is simply the foreground region. The noise is also an aspect

$$\mathbf{D} = \mathcal{P}_{\overline{\Omega}}(\mathbf{B}) + \mathbf{A} + \mathbf{E} \tag{4}$$

where $\mathbf{E}$ is the noise. Equation (4) shows the actual components of a video and is a strict constraint in our model.

A. Discriminative Exploration Using Sparse Representation

In most large-scale background modeling problems, the frames are highly redundant. Some of the frames already carry sufficient background information and are more discriminative than other frames. In this section, we refine frame sequence $\mathbf{D}$ and obtain a new informative set $\widetilde{\mathbf{D}}$, which is composed of the selected discriminative frames.

We use sparse representation to explore the discriminative frames by solving the maximum linearly independent group of video frames. The sparse representation process, which is robust to noise [34], is based on the video content. Once a frame is represented by other frames, its content is no longer discriminative. In a real-life video, the discriminative frames are those whose foreground objects are different in both position and appearance. Thus, we gain different background information from different discriminative frames. Once a sufficient amount of information is obtained, we can model the background.

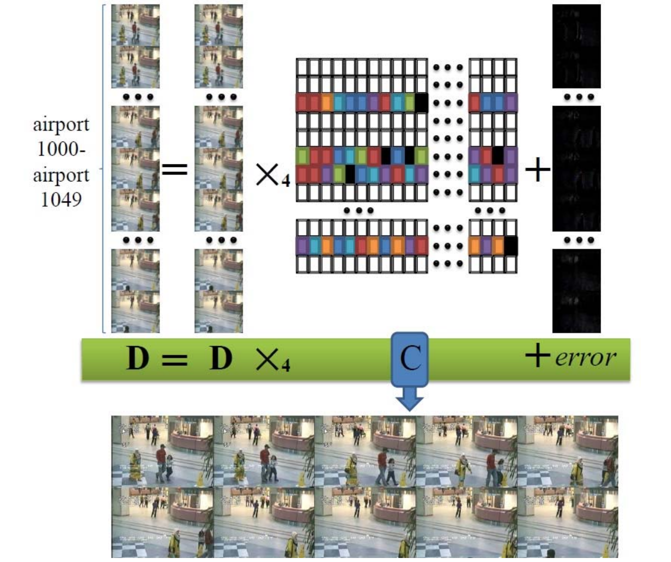


Fig. 2. Sparse representation process of a discriminative exploration. The original frame set is composed of 50 frames (airport1000–airport1049 in the *Hall* sequence from the I2R dataset), where the grid group indicates coefficient matrix C.

Now, we will introduce our sparse representation model. Foreground objects move continually in a video sequence. Two adjacent frames are usually approximately the same. Some frames can be represented through a linear combination of the remaining frames, and the other frames are usually repeated. In other words, a series of frames can represent all frames

$$\min_C \ \|\mathbf{D} - \mathbf{D}\times_4 C\|_F^2 + \lambda\|C\|_{1,2} \tag{5}$$

where $\|\cdot\|_F$ is the Frobenius norm, which equals the square root of the sum of squares of the entries of the tensor. $C \in \mathbb{R}^{N\times N}$ is a coefficient matrix, and λ is used to balance the two parts. In addition, $\|\cdot\|_{1,2}$ is the $l_{1,2}$-norm and is the sum of the l_2-norm of all the rows in C [42]. We solve this model by converting it into an equivalent problem

$$\begin{aligned}&\min_{W,C} \ \|\mathbf{D} - \mathbf{D}\times_4 W\|_F^2 + \lambda\|C\|_{1,2}\\ &\text{s.t. } W = C.\end{aligned} \tag{6}$$

This problem is the standard augmented Lagrange formulation and can be solved using the alternating direction multipliers (ADMs) method [43], [44].

The jth row of C records the coefficients of the jth frame to represent other frames, and the jth column of C records the coefficients of other frames to represent the jth frame. We can then deduce the role of each frame by observing the corresponding row in C. The frames whose coefficients are all zero are regarded as redundant, and the nonzero rows in C correspond to discriminative frames. A new set $\widetilde{\mathbf{D}}$ is formed to contain all discriminative frames.

Fig. 2 shows our sparse representation process. A video sequence equals a sparse linear combination of itself

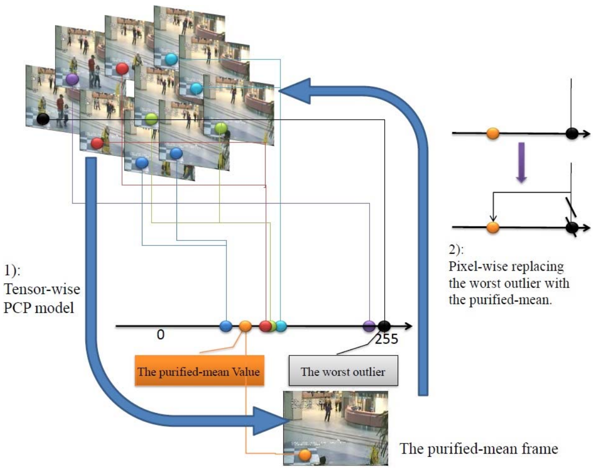


Fig. 3. Cyclic iteration process in an iteration.

plus errors. The color rows in the grid group indicate the nonzero rows in C. The corresponding frames of the nonzero rows in C are selected, i.e., the 10 frames in Fig. 2, which contain almost all background information of the 50 frames.

For most practical problems, the frames selected by (5) are suitable for background modeling. However, some abnormal behaviors and serious noise pollution of the video may lead to more complicated relations among frames. For example, in Fig. 2, if the woman in yellow jumps from left to right, more frames will be considered discriminative. However, 10 frames are sufficient for a background model. In this case, updating $\widetilde{\mathbf{D}}$ is essential. We do this based on coefficient matrix C, from which we can measure the similarity between two frames. If frame $\mathbf{I}_a$ resembles frame $\mathbf{I}_b$, the coefficient of frame $\mathbf{I}_a$ is close to 1 in representing frame $\mathbf{I}_b$; otherwise, it is far from 1. First, we find the frame that resembles all other frames in $\widetilde{\mathbf{D}}$ the most. This is the first reselected frame. Next, we choose the last similar frame of the first reselected frame. Then, every time we choose a new frame, it is the last frame that all of the previously selected frames resemble. Eventually, we choose a new frame set in which the similarities among the members are low. This set is the updated $\widetilde{\mathbf{D}}$. The appropriate number of $\widetilde{\mathbf{D}}$ will be explored experimentally.

After the sparse representation of the video, we refine the original video $\mathbf{D}$ and form a new selected discriminative frame set $\widetilde{\mathbf{D}}$. Assume that there are $\widetilde{N}$ frames in $\widetilde{\mathbf{D}} : \widetilde{\mathbf{D}} \in \mathbb{R}^{m \times n \times 3 \times \widetilde{N}}$, where $\widetilde{N} \ll N$.

B. Background Extraction Using Cyclic Iteration Process

In this section, we describe the design of a background extraction using a cyclic iteration (the outer loop in SOIR). This process is shown in Fig. 3. In each iteration, we use a PCP model to solve the purified-mean frame from the selected discriminative frame set $\widetilde{\mathbf{D}}$, and in a pixel-wise outlier removal strategy, we use the purified-mean frame to ameliorate the selected discriminative frame set $\widetilde{\mathbf{D}}$ conversely. The iteration will continue until the purified-mean frames converge to a fixed frame, which is the background.

1) Tensor-Wise PCP Model: The tensor model is used to calculate the purified mean of the selected discriminative frames. It is the mean of the frames initially, but moves slightly away during the denoising process. Following the idea of ADM, we solve this tensor model through an iterative approximation (the inner loop in SOIR). The solution is limited to an $\mathcal{R}^{(4)}$ space.

The purified-mean frame, denoted by $\mathbf{B}^* : \mathbf{B}^* \in R^{m \times n \times 3}$, is the optimal background of the current frame set $\widetilde{\mathbf{D}}$. The real backgrounds of different frames are the same, i.e., $\mathbf{B}_{\mathbf{I}_i} = \mathbf{B}_{\mathbf{I}_j} = \mathbf{B}^*, \forall i \neq j$, or

$$\widetilde{\mathbf{B}} = \mathbf{B}^* \times_4 \widetilde{Z} \tag{7}$$

where $\widetilde{Z} = (1, 1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^{\widetilde{N} \times 1}$ is a first-order matrix. $\mathbf{B}^*$ is regarded as a fourth-order tensor, i.e., $\mathbf{B}^* \in \mathbb{R}^{m \times n \times 3 \times 1}$.

Constraint (7) in fact insists that the background is rank-1. Just like the vectorizing process in [31], [35], and [39], we transform the frames into gray-scales and combine mode-1 and mode-2 of each frame into a single mode. This means that the operator $vectorize(\cdot)$ reduces the dimension of high-dimensional data. After the vectorizing process, a video tensor is transformed into a matrix, and a frame tensor is transformed into a vector. The vectorized formulation of (7) is then

$$\text{vectorize}(\widetilde{\mathbf{B}}) = \text{vectorize}(\mathbf{B}^*) \times_2 \widetilde{Z} \tag{8}$$

where $vectorize(\widetilde{\mathbf{B}}) \in R^{p \times \widetilde{N}}$ is a matrix, $vectorize(\mathbf{B}^*) \in R^p$ is a vector, and the 2-mode product ($\times_2$) is the outer product of the vectors. Thus, the equation reduces to the standard definition of rank-1.

To solve (7), we consider a subspace of $\mathcal{R}_4$. We denote this subspace by $\mathcal{R}^{(4)}$. All tensors in $\mathcal{R}^{(4)}$ are fourth-order tensors, and for each tensor $\mathbf{X}$ in this space, element wise, we have $\mathbf{X}_{ijka} = \mathbf{X}_{ijkb}, \forall a \neq b$. Thus, $\mathbf{B}^*$ must lie within this space. $\mathcal{R}^{(4)}$ is convex, and it is therefore easy to solve (7).

Lemma 1: Given tensor $\mathbf{B}$, the solution to the problem

$$\min_{\mathbf{B}^*} \|\widetilde{\mathbf{B}} - \mathbf{B}^* \times_4 \widetilde{Z}\|_F^2 \tag{9}$$

is $\mathbf{B}^* = (\sum_{l=1}^{\widetilde{N}} \widetilde{\mathbf{B}}_{\widetilde{\mathbf{I}}_l})/\widetilde{N}$.

Proof: $\|\widetilde{\mathbf{B}} - \mathbf{B}^* \times_4 \widetilde{Z}\|_F^2 = \sum_{l=1}^{\widetilde{N}} \|\widetilde{\mathbf{B}}_{\widetilde{\mathbf{I}}_l} - \mathbf{B}^*\|_F^2 = \widetilde{N}\|\mathbf{B}^* - (\sum_{l=1}^{\widetilde{N}} \widetilde{\mathbf{B}}_{\widetilde{\mathbf{I}}_l})/\widetilde{N}\|_F^2 + const$, where $const$ and $\widetilde{N}$ indicate constants. This completes the proof. ■

To model the background, we want the static video content. We therefore minimize the changing part to group more information into the background. In addition, we consider strict constraints (4) and (7) and give our model

$$\begin{aligned} \min_{\widetilde{\mathbf{B}}, \widetilde{\mathbf{A}}, \widetilde{\mathbf{E}}} \quad & \|\widetilde{\mathbf{A}} + \widetilde{\mathbf{E}} - \mathcal{P}_{\Omega}(\widetilde{\mathbf{B}})\|_1 \\ \text{s.t.} \quad & \widetilde{\mathbf{D}} = \mathcal{P}_{\overline{\Omega}}(\widetilde{\mathbf{B}}) + \widetilde{\mathbf{A}} + \widetilde{\mathbf{E}} \\ & \widetilde{\mathbf{B}} = \mathbf{B}^* \times_4 \widetilde{Z}. \end{aligned} \tag{10}$$

In the foreground region, we minimize the number of nonzero elements in $\widetilde{\mathbf{A}} + \widetilde{\mathbf{E}} - \widetilde{\mathbf{B}}$; otherwise, such pixels can be considered as background if $\widetilde{\mathbf{A}} + \widetilde{\mathbf{E}} - \widetilde{\mathbf{B}} = 0$. Outside the foreground region, we minimize the noise $\widetilde{\mathbf{E}}$. Benefitting from the definition of the l_1-norm, we arrange the two regions into a single formula, i.e., the objective function in (10).

Model (10) is a PCP model. To solve this model, we first arrange it. The constraint $\widetilde{\mathbf{B}} = \mathbf{B}^* \times_4 \widetilde{Z}$ cannot be transformed into a single variable linear equation. We therefore use it as a correction term. In addition, we denote all nonbackground parts by $\widetilde{\mathbf{S}} : \widetilde{\mathbf{S}} = \widetilde{\mathbf{A}} + \widetilde{\mathbf{E}} - \mathcal{P}_{\Omega}(\widetilde{\mathbf{B}})$. We then obtain

$$\min_{\widetilde{\mathbf{B}},\widetilde{\mathbf{S}}} \|\widetilde{\mathbf{S}}\|_1, \quad \text{s.t.} \quad \widetilde{\mathbf{D}} = \widetilde{\mathbf{B}} + \widetilde{\mathbf{S}}. \tag{11}$$

Model (11) can be solved by iteration [43] (the inner loop)

$$\begin{cases} \widetilde{\mathbf{S}}^{k+1} = \mathcal{T}_{\frac{1}{\mu}}(\widetilde{\mathbf{D}} + \dfrac{\widetilde{\Lambda}^k}{\mu} - \widetilde{\mathbf{B}}^k) \\ \widetilde{\mathbf{B}}^{k+1} = \widetilde{\mathbf{D}} - \widetilde{\mathbf{S}}^{k+1} \\ \widetilde{\Lambda}^{k+1} = \widetilde{\Lambda}^k - \mu(\widetilde{\mathbf{D}} - \widetilde{\mathbf{B}}^{k+1} - \widetilde{\mathbf{S}}^{k+1}) \end{cases} \tag{12}$$

where $\mu > 0$ is a step-length parameter and $\widetilde{\Lambda}$ is the Lagrange multiplier. In addition, $\mathcal{T}_{(1/\mu)}(\cdot)$ is a soft-threshold operator [43], [44]. For an arbitrary tensor $\mathbf{X} \in \mathbb{R}^{K_1 \times \cdots \times K_N}$, element wise, the operator is given by

$$\mathcal{T}_{\frac{1}{\mu}}(\mathbf{X})_{k_1 \ldots k_N} = \max\left(\mathbf{X}_{k_1 \ldots k_N} - \frac{1}{\mu}, 0\right) + \min\left(\mathbf{X}_{k_1 \ldots k_N} + \frac{1}{\mu}, 0\right). \tag{13}$$

Here, we consider the correction term. The tensor $\widetilde{\mathbf{B}}$ must lie in the $\mathcal{R}^{(4)}$ space. Once we obtain a new $\widetilde{\mathbf{B}}$ in (12), we project it into the $\mathcal{R}^{(4)}$ space and use its vertical projection to replace itself. Thus, the updated formula of $\widetilde{\mathbf{B}}^{k+1}$ in (12) is replaced by

$$\widetilde{\mathbf{B}}^{k+1} = \mathbf{B}^{*k+1}, \; (\mathbf{B}^{*k+1} : \widetilde{\mathbf{D}} - \widetilde{\mathbf{S}}^{k+1} = \mathbf{B}^{*k+1} \times_4 \widetilde{Z}). \tag{14}$$

The result of the inner loop is the optimal background $\mathbf{B}^*$ of the current frame set $\widetilde{\mathbf{D}}$.

2) Pixel-Wise Strategy: Once we obtain the purified-mean frame $\mathbf{B}^*$ of the current frame set $\widetilde{\mathbf{D}}$, we use $\mathbf{B}^*$ to renew the current frame set $\widetilde{\mathbf{D}}$ conversely. This is a pixel-wise method, called outlier removal strategy. We repeat this strategy for all the pixels in each frame. As an example, we then take the pixel (x, y, z) $(x = 1, \ldots, m; y = 1, \ldots, n; z = 1, 2, 3)$.

In Fig. 3, we have labeled pixel (x, y, z) in all the frames with solid color points. Different colors indicate different pixel values. When we locate these values in the axis, we find that most of the values gather into a cluster, and others do not. The outliers are the pixel values of the nonbackground pixels. The purified-mean value and worst outlier are also marked in the figure. The worst outlier is the value that is farthest away from the purified-mean value.

The ground truth of the background is inside the cluster. The purified-mean value is much closer to the ground truth than the worst outlier. Thus, the extraction performance will be improved if we use the purified-mean value to replace the worst outlier. As shown in Fig. 3, once we extract a purified-mean frame $\mathbf{B}^*$, we continue replacing the worst outlier with the purified-mean value for each pixel. Here, replacing the worst outlier means deleting the worst value and returning the purified-mean value to the frame. Thus, frame set $\widetilde{\mathbf{D}}$ is renewed after the replacement is conducted for all pixels.

C. Algorithm Formulation

The methods in Sections III-A and III-B combine to form our SOIR algorithm. In this algorithm, the sparse representation model is an important aspect. It returns the selected discriminative frames that carry sufficient background information. The cyclic iteration process is the main aspect, which extracts the background of the video. The convergence condition of the algorithm is $\|\widetilde{\mathbf{B}}^{k+1} - \widetilde{\mathbf{B}}^k\| / \|\widetilde{\mathbf{B}}^k\| \leq 1e - 3$.

Algorithm 1 SOIR Algorithm

```
Input: $\mathbf{D} \in \mathbb{R}^{m \times n \times 3 \times N}$.
Output: $\mathbf{B}^*$.
1:sparse representation:
    get $\widetilde{\mathbf{D}}$, by solving:
        $\min_C \|\mathbf{D} - \mathbf{D} \times_4 C\|_F^2 + \lambda \|C\|_{1,2}$.
2:cyclic iteration:
  while not converged do(outer loop) :
  (1): update $\mathbf{B}^*$, by:
  while not converged do(inner loop) :
      $\widetilde{\mathbf{S}}^{k+1} = \mathcal{T}_{\frac{1}{\mu}}(\widetilde{\mathbf{D}} + \frac{\widetilde{\Lambda}^k}{\mu} - \widetilde{\mathbf{B}}^k)$;
      $\widetilde{\mathbf{B}}^{k+1} = \mathbf{B}^{*k+1}$, where
          $\mathbf{B}^{*k+1} = (\sum_{l=1}^{\widetilde{N}} (\widetilde{\mathbf{D}}_{\widetilde{\mathbf{I}}_l} - \widetilde{\mathbf{S}}_{\widetilde{\mathbf{I}}_l}^{k+1})) / \widetilde{N}$;
      $\widetilde{\Lambda}^{k+1} = \widetilde{\Lambda}^k - \mu(\widetilde{\mathbf{D}} - \widetilde{\mathbf{B}}^{k+1} - \widetilde{\mathbf{S}}^{k+1})$.
  end while.
  (2): update $\widetilde{\mathbf{D}}$ $(\widetilde{\mathbf{D}} = [\widetilde{\mathbf{I}}_1, \cdots, \widetilde{\mathbf{I}}_{\widetilde{N}}])$, by:
      For pixel $(x, y, z)$
          $N^* = \text{argmax}_{\widetilde{i}} |(\widetilde{\mathbf{I}}_{\widetilde{i}})_{xyz} - \mathbf{B}^*_{xyz}|$,
          then $(\widetilde{\mathbf{I}_{N^*}})_{xyz} = \mathbf{B}^*_{xyz}$.
      End
  end while.
```

D. Convergence Analysis

We will now describe the convergence of SOIR. The convergence of a sparse representation model has been well studied [45]. Thus, we focus on the convergence of the cyclic iteration process.

For an arbitrary pixel (x, y, z), there are $\widetilde{N}$ pixel values in the selected set $\widetilde{\mathbf{D}}$. In the ith outer loop iteration, the minimum and maximum of the $\widetilde{N}$ values are recorded as a_i and b_i, respectively. $\mathbf{B}^{*i}_{xyz}$ is the purified-mean of the $\widetilde{N}$ values in the last iteration, and thus, $\mathbf{B}^{*i}_{xyz} \in [a_{i-1}, b_{i-1}]$. If $\lim_{i \to \infty}(b_i - a_i) = 0$, the purified-mean $\mathbf{B}^{*i}_{xyz}$ will converge. This can be inferred from the nested intervals theorem [46]. To complete the convergence, we prove the following lemma.

Lemma 2: In the SOIR algorithm, for an arbitrary pixel (x, y, z), the minimum and maximum of the $\widetilde{N}$ pixel values in the ith iteration are recorded as a_i and b_i, respectively. we then have $\lim_{i \to \infty}(b_i - a_i) = 0$.

Proof: First, the purified-mean value $\widetilde{\mathbf{B}}^{i+1}_{xyz}$ is inside a subinterval of the interval $[a_i, b_i]$, because the value is simply around the mean of all the $\widetilde{N}$ values in the last iteration. Otherwise, if the mean of the $\widetilde{N}$ values is close to either their minimum or maximum, we can conclude that the $\widetilde{N}$ values are already close to each other [46].

Second, after $\widetilde{N}$ iterations, we record the minimum and the maximum of all the $\widetilde{N}$ purified-mean values $(\widetilde{\mathbf{B}}^i_{xyz}, \; i = 1, 2, \ldots, \widetilde{N})$ as c and d, respectively.

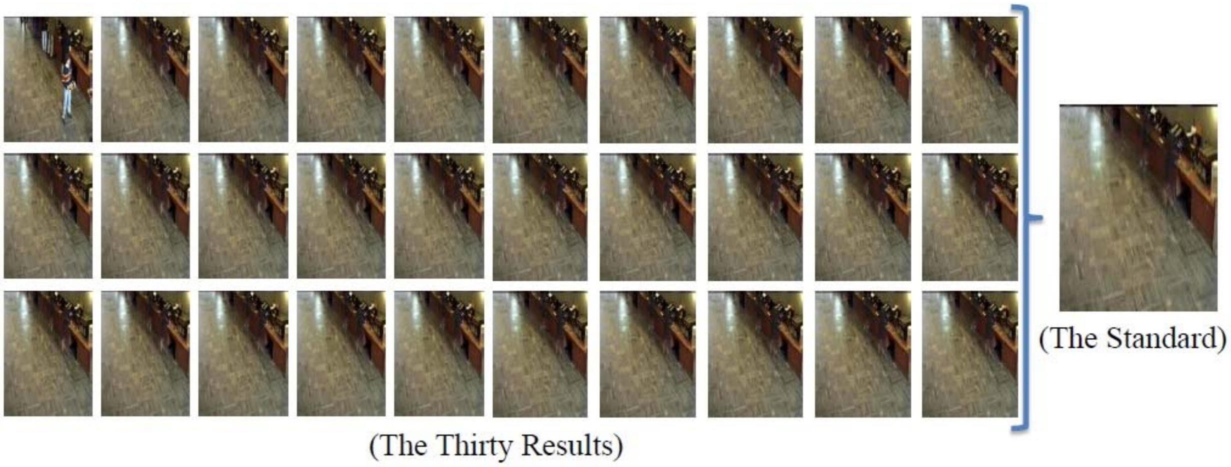


Fig. 4. Background extracting results with the size of the selected set varying from 1 to 30. The standard background is extracted when using all the 300 frames as the selected set.

Then, we have $[a_{i+\widetilde{N}}, b_{i+\widetilde{N}}] \subseteq [c, d]$, because the worst outlier is replaced by the purified-mean value in each iteration, and all of the $\widetilde{N}$ purified-mean values are in the interval $[c, d]$. The subinterval $[c, d]$ is shorter than the interval $[a_i, b_i]$. In other words, there is a constant ratio $\gamma : \gamma < 1$, subject to $(b_{i+\widetilde{N}} - a_{i+\widetilde{N}}) \leq \gamma(b_i - a_i)$.

Finally, we assume that the original minimum and maximum of the $\widetilde{N}$ values are a and b, respectively. Then $(b_{1+\widetilde{N}} - a_{1+\widetilde{N}}) \leq \gamma(b - a), (b_{1+2\widetilde{N}} - a_{1+2\widetilde{N}}) \leq \gamma^2 (b - a), \ldots, (b_{1+n\widetilde{N}} - a_{1+n\widetilde{N}}) \leq \gamma^n(b - a), \ldots$. We know that $(b - a)$ is a constant, and γ^n is close to zero when n is large. We then have $\lim_{i\to\infty}(b_i - a_i) = 0$, which completes the proof. ■

We have to point out that the derived solution may not be the ground truth and is influenced by the property of the video. The experiments in Section V will show that the solution is pretty close to the ground truth if the video quality is not too poor.

IV. Foreground Region Detection

Having computed the background tensor, as described in Section III, we then detect the foreground region of the video.

Background subtraction is a common method for detecting the foreground region. We denote the result of the subtraction by $\mathbf{F}_{\mathbf{I}_k} : \mathbf{F}_{\mathbf{I}_k} = \mathbf{I}_k - \mathbf{B}_{\mathbf{I}_k}$, where $\mathbf{I}_k = \mathcal{P}_{\overline{\mathbf{\Omega}}}(\mathbf{B}_{\mathbf{I}_k}) + \mathbf{A}_{\mathbf{I}_k} + \mathbf{E}_{\mathbf{I}_k}$ and $\mathbf{B}_{\mathbf{I}_k} = \mathbf{B}^*$. We found that the residual background exists only in the foreground region, and outside this region nothing but noise exists, that is

$$\mathbf{F}_{\mathbf{I}_k} = \begin{cases} \mathbf{A}_{\mathbf{I}_k} + \mathbf{E}_{\mathbf{I}_k} - \mathbf{B}_{\mathbf{I}_k}, & \text{inside the region } \mathbf{\Omega} \\ \mathbf{E}_{\mathbf{I}_k}, & \text{outside the region } \mathbf{\Omega}. \end{cases} \quad (15)$$

From (15), we can conclude that the background subtraction method works depending on the properties of $\mathbf{A}_{\mathbf{I}_k} - \mathbf{B}_{\mathbf{I}_k}$ and $\mathbf{E}_{\mathbf{I}_k}$, and the relationship between them. If the distribution of $\mathbf{A}_{\mathbf{I}_k} - \mathbf{B}_{\mathbf{I}_k}$ is different from that of $\mathbf{E}_{\mathbf{I}_k}$, the background subtraction will be an impactful way to detect the foreground.

We next explore the foreground region $\mathbf{\Omega}$ for an arbitrarily given image $\mathbf{I}_k$ from the original frame set $\mathbf{D}$. To simplify this problem, we transform the color frame into gray one (I_k). We model the region using MRF, following [33], [47], and [48].

First, we set up a matrix O to represent the foreground region $\mathbf{\Omega}$

$$O_{ij} = \begin{cases} 1, & (\mathbf{A}_{I_k})_{ij} \neq 0 \\ 0, & (\mathbf{A}_{I_k})_{ij} = 0. \end{cases} \quad (16)$$

The energy of $\mathbf{\Omega}$ can then be obtained using the Ising model [47]

$$\sum_{i,j} \lambda_a * O_{ij} + \sum_{i,j,x,y:|i-x|+|j-y|\leq 1} \lambda_b * |O_{ij} - O_{xy}| \quad (17)$$

where λ_a and λ_b are two positive parameters that penalize $O_{ij} = 1$ and $|O_{ij} - O_{xy}| = 1$, respectively.

Clearly, if we simply minimize the energy of the foreground region $\mathbf{\Omega}$, it will converge to an empty set, i.e., $O = 0$. To avoid this, an important component of the objective function is $\mathcal{P}_{\overline{\mathbf{\Omega}}}(\mathbf{F}_{I_k})$. In addition, the nonzero elements outside the foreground region should also be minimized. Thus, we have the following foreground detection model:

$$\min_{O_{ij}} \sum_{O_{ij}=0} \frac{1}{2}((I_k)_{ij} - (\mathbf{B}_{I_k})_{ij})^2 + \sum_{i,j} \lambda_a * O_{ij} + \sum_{i,j,x,y:|i-x|+|j-y|\leq 1} \lambda_b * |O_{ij} - O_{xy}|. \quad (18)$$

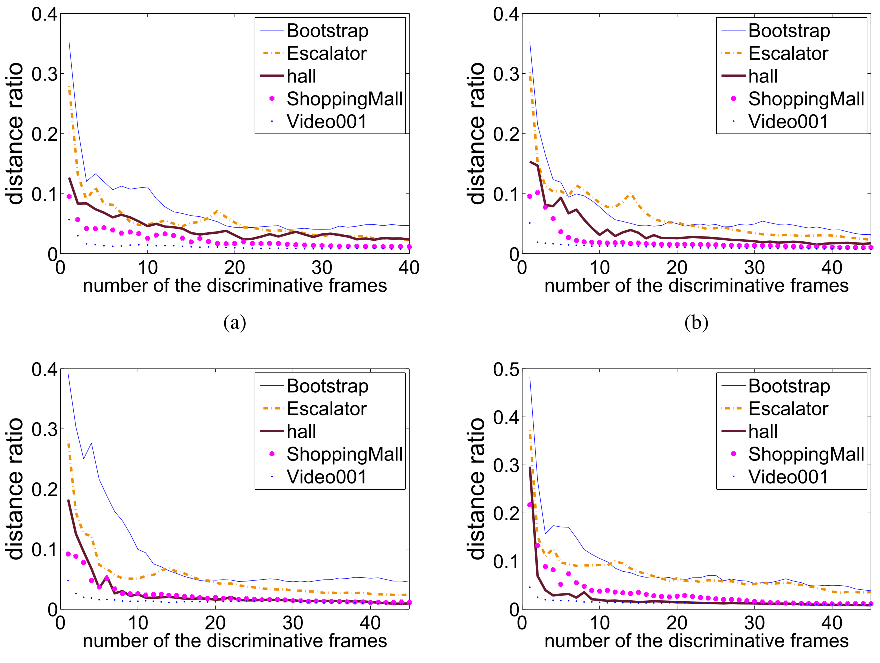
Model (18) can be rearranged as

$$\min_{O_{ij}} \sum_{ij} \left(\lambda_a - \frac{1}{2}((I_k)_{ij} - (\mathbf{B}_{I_k})_{ij})^2\right) * O_{ij} + \lambda_b * \|A \times_3 O\|_1 \quad (19)$$

where A is a projection tensor. The constant part of (18) is omitted because it is insignificant in an optimization problem. Model (19) is the standard form of a first-order MRF and can be solved exactly using graph cuts [49].

V. Experimental Analysis

In this section, we describe the performance evaluation of our SOIR algorithm. To evaluate the algorithm, we will



(a)

(b)

(c)

(d)

Fig. 5. Relationship between the number and the performance (2). (a) Original frame number is 450. (b) Original frame number is 600. (c) Original frame number is 750. (d) Original frame number is 900.

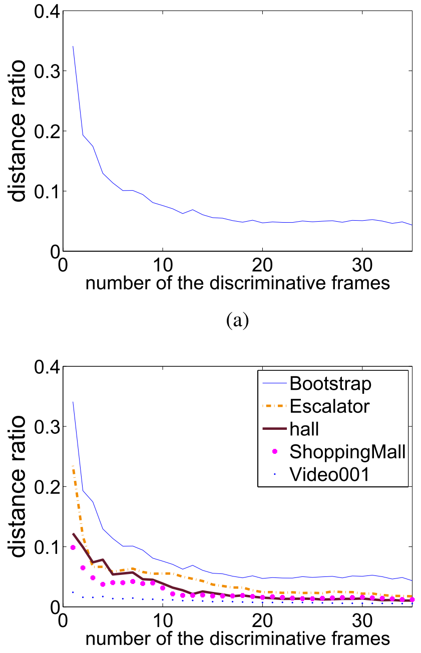
explore the appropriate number of discriminative frames and test the performance and time consumption of the algorithm. The experiments are conducted on real sequences from public datasets, such as the Institute for Infocomm Research (I2R) [24], flowerwall [50], Stuttgart Artificial Background Subtraction (SABS) [51], and Background Models Challenge (BMC) datasets [52]. In addition, some public video sequences from the Internet are also included in our experiments. All experiments are conducted and timed in MATLAB R2010a on a PC with a 3.20-GHz Intel(R) Core(TM) CPU and 4 GB of RAM.

A. Number of Discriminative Frames

A major aspect of this paper is utilization of sparse representation to reduce the size of the video. In this section, we explore the appropriate number of discriminative frames.

First, we provide the details of our experiment on the Bootstrap sequence of the I2R dataset. A scene from this video takes place in front of a buffet. We use the first 300 frames in the sequence as our original frame set $\mathbf{D}$ and measure the performance of the SOIR algorithm when the number of frames of the selected discriminative frame set $\widetilde{\mathbf{D}}$ varies from 1 to 30. For comparison, we need a standard background, which we use all the 300 frames to extract.

The results are shown in Fig. 4. In the figure, we can see that most of the extracted backgrounds are quite similar to the standard, even when the number of frames is small. However, it is a little disappointing that the counter is not recovered exactly, even in our standard background. The two small fuzzy areas are the spaces just in front of the buffet, where people are continuously standing and taking the meal in nearly every frame. We measure the relationship between the rate of convergence and the number of discriminative frames, the results of which are shown in Fig. 6(a).



(a)

(b)

Fig. 6. Relationship between the number and the performance (1). Distance ratio is to divide the distance between the result and the standard by that between the standard and the original of coordinate. (a) Distance ratio of "Bootstrap"; (b) Distance ratios of various video sequences.

Next, we repeat the operations on some additional video clips, i.e., the *Escalator*, *Hall*, and *ShoppingMall* sequences in the I2R dataset, and a real video sequence in the BMC dataset. The results of each video sequences (including Bootstrap) are shown in Fig. 6(b).

We can observe from Fig. 6 that the performance is not very high when the number of frames is small.

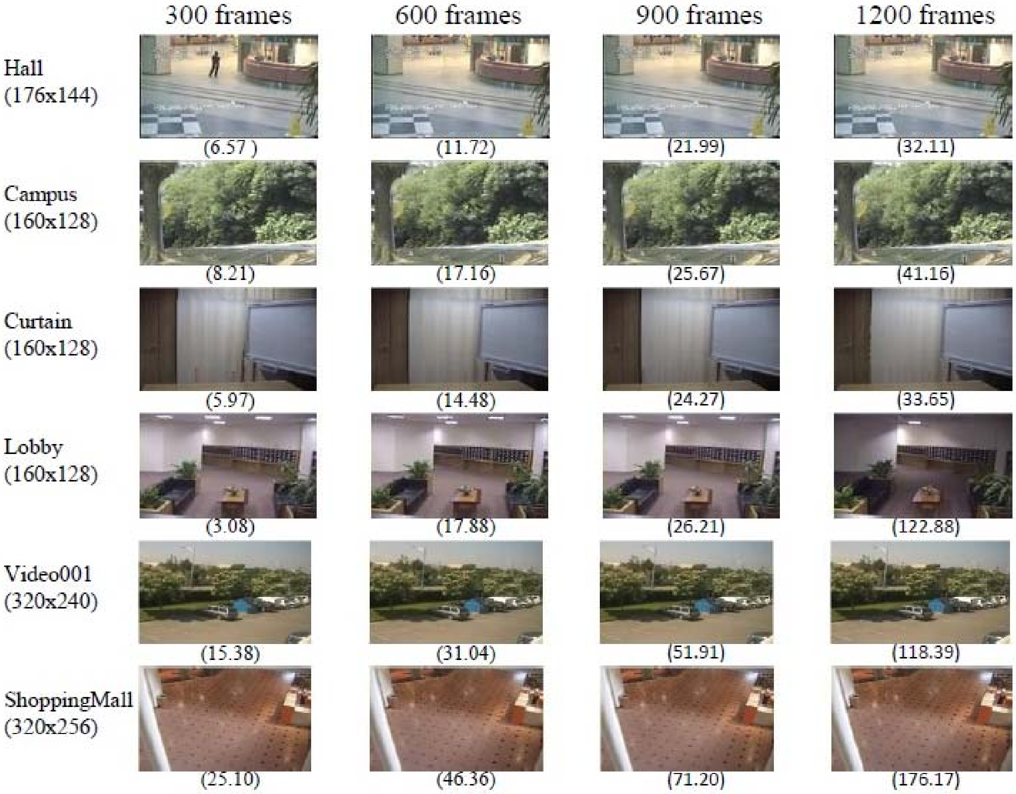


Fig. 7. Experiments on different video sequences. The results are shown together with the time consumption (unit of time is seconds). The sequences are from the I2R dataset and the BMC dataset.

However, it improves as the number of frames increases. When the number of frames is larger than 20, the ratio tends to become stable. The small fluctuation of each curve is caused by the nonbackground information of each new frame; however, the influence of this weakens after the frame is processed using our algorithm. In addition, we also find that the content of the video affects the results. In the Video001 sequence, the foreground region is small. Thus, a small amount of frames already carry sufficient background information, and the curve is smooth. However, in the Bootstrap sequence, we use many more frames to deal with the changes in illumination, although the 30 results shown in Fig. 4 all look the same.

Finally, we also explore the appropriate number of discriminative frames when the original frame number is larger, the results of which are shown in Fig. 5. We can see that the curve varies when the original number of frames increases because the discriminative frames are different. However, the distance ratio tends to become stable in all experiments when the number of discriminative frames increases to around 35. The ratio will improve if we use more frames, but the effect is unremarkable. This means that 35 frames or so already carry sufficient background information in most cases. If the content of the video sequence is pretty simple, fewer frames will be required, and oppositely, more frames will be needed if the content is complex.

B. Experiments on the Time Consumption

In this section, we describe the time consumption of our model when solving tasks of different sizes. The majority of traditional methods are used for sequences with resolutions of around 150 × 150 and where the number of frames is usually around 50. When the scale of the data increases, these methods tend to become inefficient.

First, we focus on the number of frames of the sequence. We extract the background in four levels, i.e., from the first 300 frames, the first 600 frames, the first 900 frames, and the first 1200 frames. The results are shown in Fig. 7.

When dealing with sequences with larger numbers of frames than usual, our model solves the background efficiently. Hundreds of frames only cost us dozens of seconds. As the number of frames increases, the precision of the extracted background is improved. Temporary static motion is a problem that exists in most traditional background modeling methods. Once a person remains at a particular spot for a while, he may be considered a part of the background in a short video sequence. In our experiments, as the number of frames increases, the problem of temporary stay is perfectly solved, as illustrated in the results on the *Hall* sequence. We can also see that the time consumption is nonlinear with the number of frames. On the one hand, the uncertainty of a background created by the temporary stay may cost some additional time. On the other hand, the foreground content and noise also influence the time consumption.

Next, we test our model on video sequences of both low and high resolutions. We use four video sequences, the first of which is from the BMC dataset, and the other three are inter-section monitoring video sequences from a public resource. We test our model on the first 50 frames and 150 frames of each sequence. The results are shown in Fig. 8.

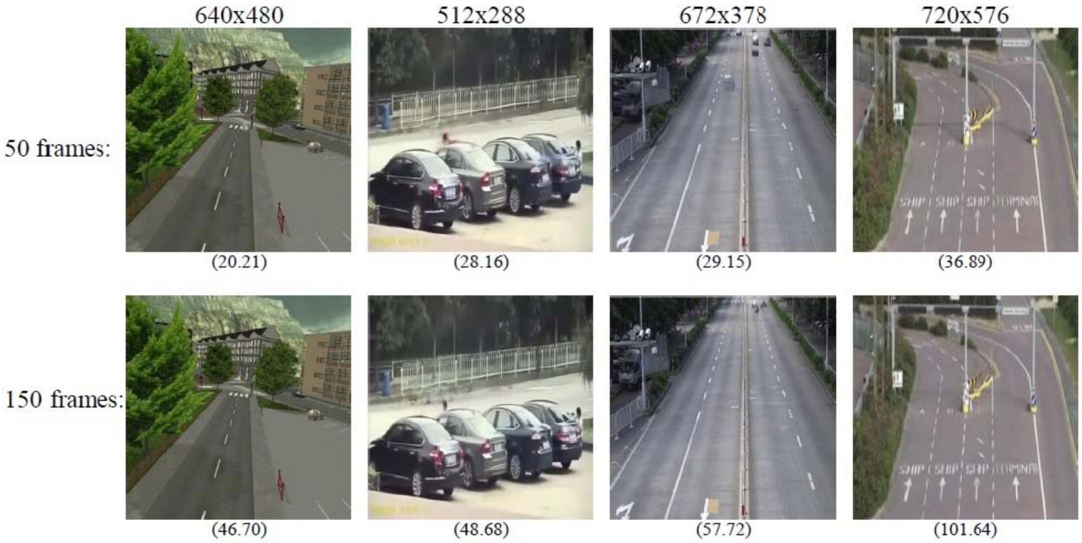


Fig. 8. Experiments on the videos whose resolutions are much higher. The results are shown together with the time consumption (unit of time is seconds). The resolutions are given out at the top of each column. The leftmost video is from the BMC dataset while the others are from the Internet.

TABLE I
TIME CONSUMPTION OF PCP, DECOLOR, SG, AND SOIR

video	number	PCP	DECOLOR	SG	SOIR
MovedObject	150	24.66	43.76	1.37	2.65
	300	53.97	74.66	2.81	6.19
	450	85.52	124.12	4.30	11.90
Bootstrap	150	61.27	186.18	1.58	6.25
	300	124.61	491.57	3.43	7.5
	450	207.56	902.26	5.43	10.59
hall	150	64.57	144.28	2.00	3.75
	300	154.71	303.98	4.47	5.94
	450	246.26	599.91	7.26	9.61
Campus	150	39.26	27.18	1.72	5.66
	300	89.01	47.13	3.63	6.94
	450	150.67	80.14	6.77	12.09

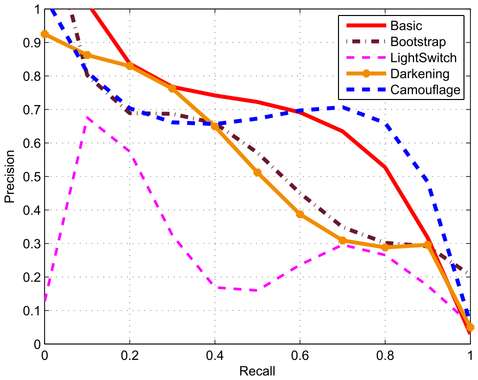


Fig. 9. Precision and recall of our method for the videos in the SABS dataset.

The man-made video from the BMC dataset consumes the least amount of time. In the later three real-life videos, the time consumption increases as the resolution of each video sequence increases. Our model spends dozens of seconds solving the high-resolution video sequences. In addition, we can also conclude from Fig. 8 that the content of the video sequence also influences the performance. In the third video, the distant cars move slowly in the fixed lens owing to the perspective, which is actually an approximation of temporary stay. We can see that 150 frames are still insufficient to resolve this phenomenon, and more frames are needed.

For comparison, we also examine the time consumption of some additional methods. Because our model is a PCP model, two PCP-based methods are included, i.e., the PCP [29] and the detecting contiguous outliers in low-rank representation (DECOLOR) [33], which perform well for small-scale problems. In addition, we also use the SG model [13], which is currently the fastest method [3].

In the experiments, we use the MovedObject and Bootstrap sequences from the flowerwall dataset, and the *Hall* and *Campus* sequences from the I2R dataset. For each video sequence, we use 150, 300, and 450 frames as the original frame set to extract the background. As shown in Table I, SG is the fastest, but its speed is maintained at the expense of accuracy, which is lower than that of most other popular methods [3].

SOIR is on average 10 times faster than other PCP-based models and can almost reach the speed of SG. This level of speed is because of the result of SOIR extracting the background from the discriminative frames, instead of the original frame set. When the scale of the data is large, the major time consumption of SOIR is in exploring these discriminative frames. Once they are obtained, however, we can model the background quickly and precisely. In the next section, we will show how the accuracy of our method is high in dealing with real-life video sequences.

C. Detecting the Foreground

1) Evaluation of Artificial Dataset: In this section, we provide the evaluation results obtained using the SABS dataset. Some approaches in [52] and [53] have been evaluated on the SABS dataset, and their recall–precision curves have been given. For a comparison of these curves, we also evaluate our algorithm on the SABS dataset and give the corresponding recall–precision curves of our algorithm.

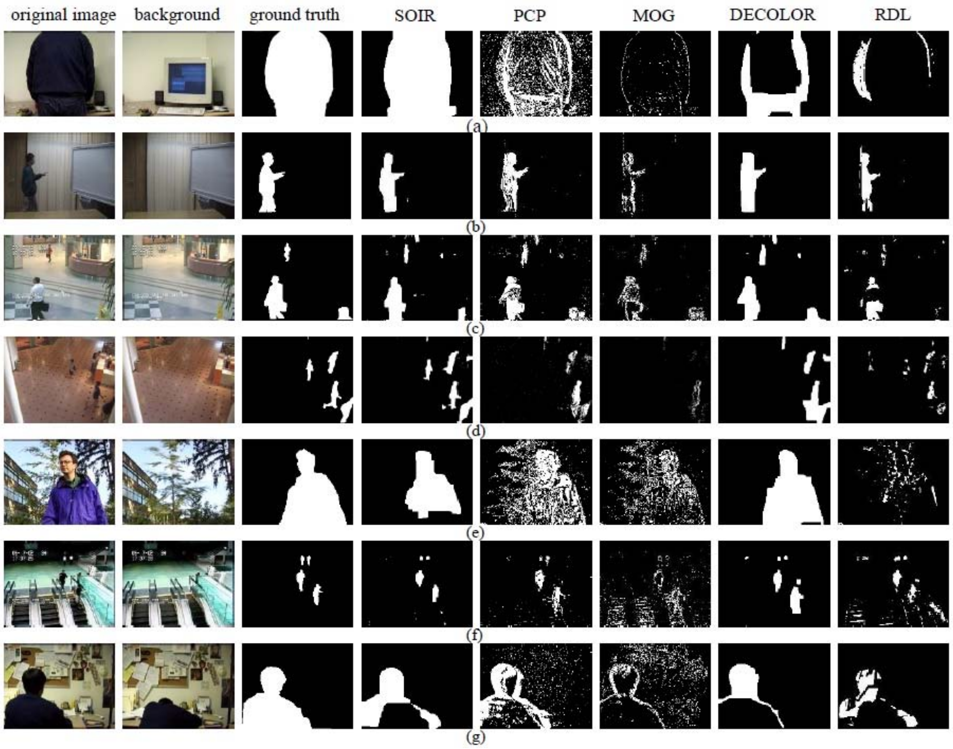


Fig. 10. Results of detecting the foreground. From left to right: original image, exacted background by SOIR, ground truth, SOIR, PCP, MOG, DECOLOR, RDL. From top to bottom: (a) *Camouflage* (b00251, flowerwall), (b) *Curtain* (Cur-1 tain22772, I2R), (c) *hall* (airport2180, I2R), (d) *ShoppingMall* (*ShoppingMall*1980, I2R), (e) *WavingTrees* (b00247, flowerwall), (f) *Escalator* (airport4595, I2R), (g) *ForegroundAperture* (b00489, flowerwall).

In Fig. 9, the curves are evaluated on different scenes in the SABS dataset. Our method performs well for most scenes, especially in the *Basic* and *Camouflage* sequences. The performance for the *LightSwitch* sequence is not very good. As shown in [52] and [53], the light switch in the video is a huge challenge for most foreground detection methods.

2) Evaluation on Real Scenes: We compare the performance of our method with those of some other studies, i.e., MOG [14], PCP [29], DECOLOR [33], and robust dictionary learning (RDL) [39]. Here, the fastest SG is not included, because its accuracy is lower than the accuracy of most popular methods [3]. Thus, we use a more complex Gaussian model, i.e., MOG. We use the video sequences from the I2R and flowerwall datasets and compare the detected foreground region with the given hand-segmented foreground region. The test frame is chosen randomly from all hand-segmented frames. To avoid the influence of a temporary stay, we use 250 frames, the last of which is the test frame.

The sequences and results are shown in Fig. 10. In the experiments, SOIR can extract the background exactly for almost all of the sequences and is robust to noise. In video sequence (g), the man remains at the same spot in all 250 frames. We can see that our algorithm is robust to noise and performs well in foreground detection, which benefits from the accurate results of the background and the MRF model. DECOLOR also performs well because it also models the foreground using the MRF model. In most sequences, the results of SOIR are better than those of DECOLOR, because the extracted backgrounds of our algorithm are more exact.

TABLE II

F-MEASURES OF THE SEQUENCES SHOWN IN FIG. 10

Sequence	SOIR	PCP	MOG	DECOLOR	RDL
(a)	**0.9737**	0.6110	0.2047	0.5669	0.1170
(b)	**0.9020**	0.7129	0.3841	0.8244	0.7326
(c)	**0.8452**	0.6986	0.5406	0.7225	0.6160
(d)	**0.8314**	0.5248	0.2498	0.6439	0.5367
(e)	0.8170	0.6046	0.4014	**0.8966**	0.3476
(f)	**0.7972**	0.5902	0.2455	0.6487	0.2399
(g)	**0.6382**	0.5104	0.1962	0.3941	0.5409

To quantitatively evaluate the performance of the different algorithms, we compute the F-measure, which is derived from the precision and recall and is computed as

$$F\text{-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{20}$$

Table II shows the F-measures of all detected foreground regions in Fig. 10. We can see that the results of SOIR are better than those of the other four methods for six sequences, i.e., Fig. 10(a)–(d), (f), and (g). However, it is a little worse than DECOLOR in the sequence in Fig. 10(e). We can also see that the performance of SOIR varies among the different video sequences. On the one hand, this is due to the result of the background extraction, which is the case for the sequence in Fig. 10(g). On the other hand, the instability of the background also affects the performance of SOIR.

VI. Conclusion

In this paper, we propose an SOIR algorithm to model the background of a video sequence. We find that a few discriminative frames are already sufficient to model the background. We use the sparse representation process to reduce the size of the video. Although it takes our algorithm some time to explore the discriminative frames, it saves much more time in modeling the background. A cyclic iteration process is proposed for background extraction. SOIR achieves both high accuracy and high speed simultaneously when dealing with real-life video sequences. In particular, SOIR has an advantage in solving large-scale tasks.

As future work, we will deal with some additional complex problems in which the background is no longer stable among different frames.

References

[1] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2011, pp. 1305–1312.

[2] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2012, pp. 864–877.

[3] T. Bouwmans, "Recent advanced statistical background modeling for foreground detection: A systematic survey," *Recent Patents Comput. Sci.*, vol. 4, no. 3, pp. 147–176, 2011.

[4] C. Kim and J.-N. Hwang, "Object-based video abstraction for video surveillance systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1128–1138, Dec. 2010.

[5] S. Li, H. Lu, and L. Zhang, "Arbitrary body segmentation in static images," *Pattern Recognit.*, vol. 45, no. 9, pp. 3402–3413, Sep. 2012.

[6] D. Park and H. Byun, "A unified approach to background adaptation and initialization in public scenes," *Pattern Recognit.*, vol. 46, no. 7, pp. 1985–1997, Jul. 2013.

[7] W. Wang, J. Yang, and W. Gao, "Modeling background and segmenting moving objects from compressed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 5, pp. 670–681, May 2008.

[8] M. Van Droogenbroeck and O. Paquot, "Background subtraction: Experiments and improvements for ViBe," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. Workshop*, Providence, RI, USA, Jun. 2012, pp. 32–37.

[9] V. Mahadevan and N. Vasconcelos, "Background subtraction in highly dynamic scenes," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, Jun. 2008, pp. 1–6.

[10] Y.-T. Chen, C.-S. Chen, C.-R. Huang, and Y.-P. Hung, "Efficient hierarchical method for background subtraction," *Pattern Recognit.*, vol. 40, no. 10, pp. 2706–2715, Oct. 2007.

[11] S.-C. Wang, T.-F. Su, and S.-H. Lai, "Detecting moving objects from dynamic background with shadow removal," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Prague, Czech Republic, May 2011, pp. 925–928.

[12] A. Ulges and T. M. Breuel, "A local discriminative model for background subtraction," in *Pattern Recognition*. Berlin, Germany: Springer-Verlag, Jun. 2008, pp. 507–516.

[13] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[14] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for realtime tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Fort Collins, CO, USA, Jun. 1999, pp. 246–252.

[15] D.-S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.

[16] J. K. Suhr, H. G. Jung, G. Li, and J. Kim, "Mixture of Gaussians-based background subtraction for Bayer-pattern image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 3, pp. 365–370, Mar. 2011.

[17] H.-H. Lin, J.-H. Chuang, and T.-L. Liu, "Regularized background adaptation: A novel learning rate control scheme for Gaussian mixture modeling," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 822–836, Mar. 2011.

[18] J.-M. Guo, Y.-F. Liu, C.-H. Hsia, and C.-S. Hsu, "Hierarchical method for foreground detection using codebook model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 6, pp. 804–815, Jun. 2011.

[19] A. Zaharescu and M. Jamieson, "Multi-scale multi-feature codebook-based background subtraction," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, Barcelona, Spain, Nov. 2011, pp. 1753–1760.

[20] A. M. Hamad and N. Tsumura, "Background subtraction based on time-series clustering and statistical modeling," *Opt. Rev.*, vol. 19, no. 2, pp. 110–120, Mar. 2012.

[21] A. M. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric model for background subtraction," in *Proc. 6th Eur. Conf. Comput. Vis.*, Jun. 2000, pp. 751–767.

[22] E. Learned-Miller, M. Narayana, and A. Hanson, "Background modeling using adaptive pixelwise kernel variances in a hybrid feature space," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 2104–2111.

[23] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[24] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tians, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.

[25] L. Zhang, W. Dong, X. Wu, and G. Shi, "Spatial-temporal color video reconstruction from noisy CFA sequence," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 6, pp. 838–847, Jun. 2010.

[26] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikainen, and S. Z. Li, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 1301–1306.

[27] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, Aug. 2000.

[28] Z. Zhou, X. Li, J. Wright, E. J. Candès, and Y. Ma, "Stable principal component pursuit," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, USA, Jun. 2010, pp. 1518–1522.

[29] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, May 2011, Art. ID 11.

[30] G. Tang and A. Nehorai, "Robust principal component analysis based on low-rank and block-sparse matrix decomposition," in *Proc. 45th IEEE Annu. Conf. Inf. Sci. Syst.*, Baltimore, MD, USA, Mar. 2011, pp. 1–5.

[31] C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Foreground detection based on low-rank and block-sparse matrix decomposition," in *Proc. 19th IEEE Int. Conf. Image Process.*, Orlando, FL, USA, Oct. 2012, pp. 1225–1228.

[32] Y. Xu, D. Zhang, and J.-Y. Yang, "A two-phase test sample sparse representation method for use with face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1255–1262, Sep. 2011.

[33] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, Mar. 2013.

[34] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[35] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 1600–1607.

[36] W. Zuo, D. Meng, L. Zhang, X. Feng, and D. Zhang, "A generalized iterated shrinkage algorithm for non-convex sparse coding," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, Australia, Dec. 2013, pp. 217–224.

[37] J. Yang, Y. Wang, W. Xu, and Q. Dai, "Image and video denoising using adaptive dual-tree discrete wavelet packets," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 642–655, May 2009.

[38] R. Sivalingam, A. D'Souza, M. Bazakos, R. Miezianko, V. Morellas, and N. Papanikolopoulos, "Dictionary learning for robust background modeling," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 4234–4239.

[39] C. Zhao, X. Wang, and W.-K. Cham, "Background subtraction via robust dictionary learning," *EURASIP J. Image Video Process.*, vol. 2011, pp. 1–12, Feb. 2011, no. 972961.

[40] B. W. Bader and T. G. Kolda, "Algorithm 862: MATLAB tensor classes for fast algorithm prototyping," *ACM Trans. Math. Softw.*, vol. 32, no. 4, pp. 635–653, Dec. 2006.

[41] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[42] J. A. Tropp, “Algorithms for simultaneous sparse approximation. Part II: Convex relaxation,” *Signal Process.*, vol. 86, no. 3, pp. 589–602, Mar. 2006.

[43] X. Yuan and J. Yang, “Sparse and low-rank matrix decomposition via alternating direction methods,” *Pacific J. Optim.*, vol. 9, no. 1, pp. 167–180, 2013.

[44] Z. Lin, R. Liu, and Z. Su, “Linearized alternating direction method with adaptive penalty for low rank representation,” in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates, Inc., Dec. 2011.

[45] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, New York, NY, USA, 2009, pp. 689–696.

[46] R. Johnsonbaugh and W. E. Pfaffenberger, *Foundations of Mathematical Analysis*. New York, NY, USA: Dover, 2012.

[47] S. Li, *Markov Random Field Modeling in Image Analysis*. New York, NY, USA: Springer-Verlag, 2009.

[48] C.-Y. Chung and H. H. Chen, “Video object extraction via MRF-based contour tracking,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 1, pp. 149–155, Jan. 2010.

[49] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[50] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: Principles and practice of background maintenance,” in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Kerkyra, Greece, Sep. 1999, pp. 255–261.

[51] S. Brutzer, B. Hoferlin, and G. Heidemann, “Evaluation of background subtraction techniques for video surveillance,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2013, pp. 1937–1944.

[52] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequievre, “A benchmark dataset for outdoor foreground/background extraction,” in *Proc. Asian Conf. Comput. Vis. Workshop*, Daejeon, Korea, Nov. 2012, pp. 291–300.

[53] A. Shimada, H. Nagahara, and R.-I. Taniguchi, “Background modeling based on bidirectional analysis,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2013, pp. 1979–1986.

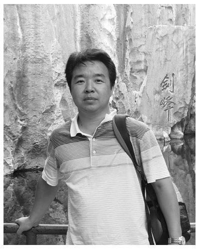
Linhao Li was born in 1989. He received the B.S. degree in applied mathematics and the M.S. degree in computational mathematics from Tianjin University, Tianjin, China, in 2012 and 2014, respectively, where he is currently working toward the Ph.D. degree with the School of Computer Science and Technology.

His research interests include low-rank matrix recovery, machine learning, background modeling and sparse coding.

Ping Wang was born in 1967. She received the B.S., M.S., and Ph.D. degrees from Tianjin University, Tianjin, China, in 1988, 1991, and 1998, respectively.

She is a Professor, Master’s Supervisor, and Ph.D. Supervisor with the Department of Mathematics, Tianjin University. Her research interests include signal and information processing, pattern recognition, and image processing.

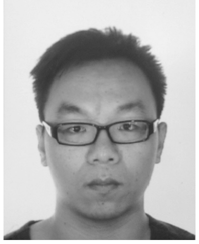
Prof. Wang is also a member of the Christ’s Commission Fellowship (E200011015M) and the Association for Computing Machinery (4157113).



Qinghua Hu (SM’13) received the B.S., M.S., and Ph.D. degrees from Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively.

He was a Post-Doctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, from 2009 to 2011. He is currently a Full Professor and the Vice Dean of the School of Computer Science and Technology with Tianjin University, Tianjin, China. He has authored over 100 journal and conference papers in the areas of granular computing-based machine learning, reasoning with uncertainty, pattern recognition, and fault diagnosis. His research interests include rough sets, granular computing, and data mining for classification and regression.

Prof. Hu was acted as the Program Committee Co-Chair of the International Conference on Rough Sets and Current Trends in Computing in 2010, the Chinese Rough Set and Soft Computing Society in 2012 and 2014, and the International Conference on Rough Sets and Knowledge Technology and the International Conference on Machine Learning and Cybernetics in 2014, and serves as a Referee for a great number of journals and conferences.



Sijia Cai was born in 1988. He received the B.S. degree in applied mathematics and the M.S. degree in computational mathematics from Tianjin University, Tianjin, China, in 2011 and 2014, respectively.

He is an Exchange Student with Hong Kong Polytechnic University, Hong Kong. His research interests include sparse optimization, multilinear analysis, image processing, and machine learning.