

Foreground-Background Separation From Video Clips via Motion-Assisted Matrix Restoration

Xinchen Ye, Jingyu Yang, Xin Sun, Kun Li, Chunping Hou, and Yao Wang, *Fellow, IEEE*

Abstract—Separation of video clips into foreground and background components is a useful and important technique, making recognition, classification and scene analysis more efficient. In this paper, we propose a motion-assisted matrix restoration (MAMR) model for foreground-background separation in video clips. In the proposed MAMR model, the backgrounds across frames are modeled by a low-rank matrix, while the foreground objects are modeled by a sparse matrix. To facilitate efficient foreground-background separation, a dense motion field is estimated for each frame, and mapped into a weighting matrix which indicates the likelihood that each pixel belongs to the background. Anchor frames are selected in the dense motion estimation to overcome the difficulty of detecting slowly-moving objects and camouflages. In addition, we extend our model to a robust MAMR model (RMAMR) against noise for practical applications. Evaluations on challenging datasets demonstrate that our method outperforms many other state-of-the-art methods, and is versatile for a wide range of surveillance videos.

Index Terms—Background segmentation/subtraction, motion detection, optical flow, matrix restoration, video surveillance.

I. INTRODUCTION

VIDEOS have become the basic representation of interesting scenes and events, and are widely used in many areas such as entertainment, public-security surveillance, healthcare. As a consequence, video analysis is of crucial importance to mine interesting information from mass data [1]–[3]. Separation of foreground and background [4]–[7] is to divide a video clip into two complementary components: the background and the foreground, which has become a useful technique for video analysis in many applications such as motion detection [8], [9], object recognition [10], and video coding [11].

For accurate foreground-background separation, there are many tough problems arising from the practical applications, for example, illumination changes: the background has intensity variations due to lighting changes [12]; camouflage: slowly-moving objects are difficult to identify, resulting in wrong

classification; noise: video signals are usually contaminated by various types of noise. Previous methods, such as Gaussian mixture model (GMM) [13], non-parametric kernel density estimation [14], and methods based on robust principle component analysis (RPCA) [15], have addressed some of these factors and made significant progress (detailed in Sec. II), but more research work is still necessary to achieve more accurate separation of foreground and background components in video clips.

In this paper, we propose a new foreground-background separation method via motion-assisted matrix restoration (MAMR). Figure 1 illustrates the work flow of our method. The main idea is to incorporate motion information into the matrix recovery framework to facilitate the separation of the foreground and the background. To this end, a dense motion field is first estimated for each frame against an anchor frame, and mapped into a weighting matrix which indicates the likelihood that each pixel belongs to the background. Anchor frames are selected in the dense motion estimation process to overcome the difficulty in detecting slowly-moving objects and camouflages. The separation problem is then formulated into a motion-assisted matrix restoration (MAMR) model with the weighting matrix. The model is solved by the alternating direction method under the augmented Lagrangian multiplier (ADM-ALM) framework. Then we estimate the foreground using our background subtraction technique. In addition, we extend our model to a robust MAMR model (RMAMR) for practical applications. Experiments show that our method achieves consistently better performance than many state-of-the-art methods on various datasets with different characteristics (e.g., motions, lighting conditions, and noise).

The rest of this paper is organized as follows. In Sec. II, we present a brief overview of related work. Sec. III presents the formulation of weighting matrix, the MAMR model, and the extended RMAMR model; we further develop the ADM-ALM algorithm to solve the proposed models in this section. Experimental results and analysis are given in Sec. IV, and conclusions are drawn in Sec. V.

II. RELATED WORK

Background extraction and foreground detection techniques can be divided into two categories: local methods and global methods. We give a brief overview for these two categories.

A. Local Methods

Local methods usually operate on each pixel individually. Some simple methods, including running Gaussian average

© 2014 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61372084 and Grant 61302059, and in part by the Tianjin Research Program of Application Foundation and Advanced Technology under Grant 12JCYBJC10300 and Grant 13JCQNJC03900.

J. Yang, X. Ye, X. Sun and C. Hou are with the School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: yjy@tju.edu.cn, yexch@tju.edu.cn, sunxin1012@gmail.com, hcp@tju.edu.cn).

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

Yao Wang is with the Polytechnic Institute of New York University, Brooklyn, NY 11201 USA (e-mail: yao@poly.edu).

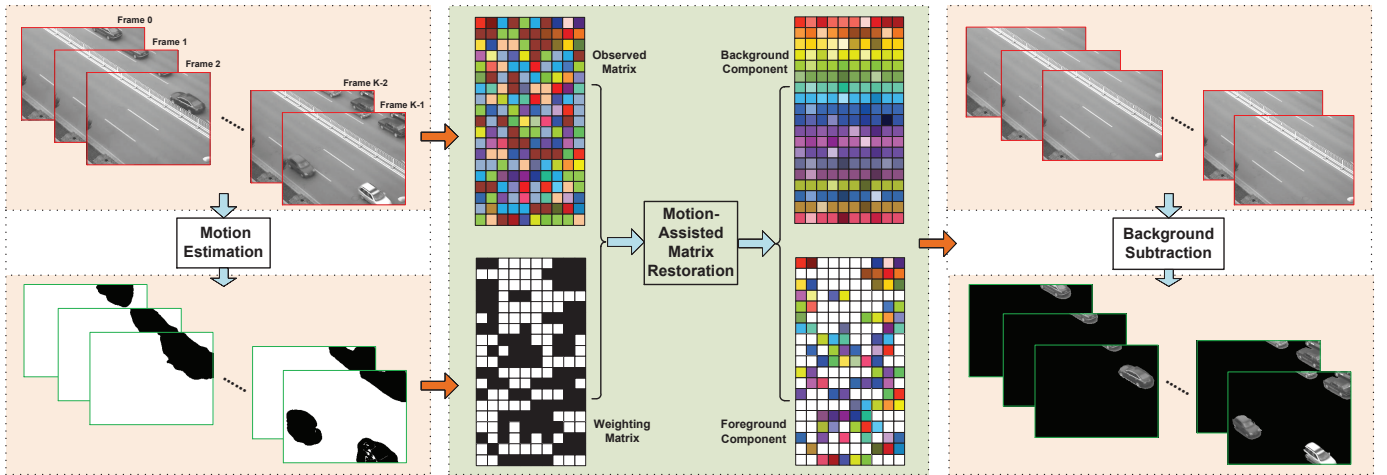


Fig. 1. The work flow of the proposed method.

[16], temporal median filtering [17], and first-order low-pass filtering [18], in some cases offer satisfactory accuracy with high processing speed, but have difficulties to deal with backgrounds with multi-modal intensity distributions. To model the multi-modality background, methods based on Gaussian mixture model (GMM) [13], [19], [20] achieve significant improvements, but are still difficult to handle challenging video clips with varying lighting conditions and/or dynamic backgrounds. The non-parametric model based on kernel density estimation [14] is more robust to rapid variations of backgrounds. ViBe [21], which is also a non-parametric model method, introduces a random strategy to update the background values. Hofmann *et al.* [22] proposed the pixel-based adaptive segmenter (PBAS) by assigning adaptive randomness parameters. Besides, Godbehere *et al.* [23] introduced a pixel-wise Bayesian segmentation algorithm that identifies foreground objects from an inferred foreground model and an estimated background. Yao *et al.* [24] introduced a robust multi-layer background subtraction technique which takes advantages of local texture features represented by local binary patterns and photometric invariant color measurements in RGB color space. Self-organization background subtraction (SOBS) [25] proposed by Maddalena *et al.* learned background motion with a self-organizing neural network, and obtains impressive detection results for scenes with gradual illumination variations. The $\Sigma - \Delta$ motion detection filter [26] is applicable to embedded systems, but compromises on the detection accuracy to some extent.

Generally, local methods enjoy the simplicity in design and implementation, but the resulting segmentation map often suffers from spatial inconsistency. Also, these techniques are sensitive to perturbations (e.g., noise, illumination variations), and yield misclassifications around boundaries between the background and foreground.

B. Global methods

In contrast to local methods, global methods exploit more spatial correlation information. Markov random field (MRF) based methods are frequently used in background extraction

for integrating spatial or spatial-temporal information. Yue *et al.* [27] presented a time dependent MRF model with multi-resolution spatiotemporal pyramids. More recently, based on fuzzy GMM and MRF, Zhao *et al.* [28] introduced the spatiotemporal constraints into the model to deal with dynamic backgrounds.

Principal component analysis (PCA), widely used in classic data analysis, is also powerful in background modeling. Seki *et al.* [29] trained a PCA for each block-volume over time, and determined the belonging (to the background or foreground) of each block by measuring its projection to the trained PCA. The eigenspace model [30] is proposed to detect moving objects. Using blocks as basic units, PCA-based methods are prone to misclassifying pixels at foreground-background boundaries.

Robust PCA (RPCA) [15], a well-known extension of PCA, is able to efficiently exploit the underlying low-rank structure in the data even in the presence of large errors or outliers. Recently, many background and foreground separation methods based on RPCA have been developed [31]–[36]. Gao *et al.* [31] introduced a two-pass RPCA combining with motion saliency estimation to detect foreground. Guyon *et al.* [32] proposed an adapted $\ell_{2,1}$ norm to model the sparse component, which satisfies the ad hoc block-sparse hypothesis. Zhou *et al.* [9] improved previous RPCA-based methods by using ℓ_0 norm instead of ℓ_1 norm to model the sparse component, and incorporating contiguity prior using MRF to make the foreground objects spatially consistent. Bouwmans *et al.* [33] presented a comprehensive review on RPCA-PCP based methods [34]–[36] for testing and ranking existing algorithms for foreground detection.

In general, many methods have been developed using the framework of sparse representation and rank minimization. However, previous methods are motion-unaware and would introduce smearing artifacts when handling slow motion and motionless foreground (camouflages). To be aware of motions, our work encodes motion information into the low-rank and sparse recovery model by a weighting matrix, which is distinct from the recent work in [9] that improves RPCA by imposing smoothness of the foreground component. The proposed

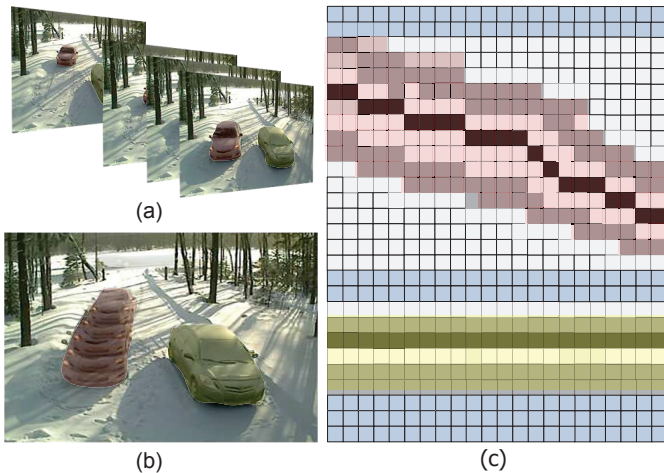


Fig. 2. Illustration of the deficiency of the RPCA-based modeling: (a) continuous frames from a video clip; (b) the overlapping of all the frames; (c) observed matrix constructed by frames in (a). For easy observation, the left car is marked by red region while the right one marked yellow. The corresponding pixels of the two cars in the observed matrix are also marked by the same colors, respectively.

method also preserves the spatial smoothness of the foreground component to some extent as the used optical flow estimator [37] has considered the smoothness of the motion field (hence the foreground). Our successful attempt might serve as a good starting point to exploit the incorporation of more complex motion models or other clues into the low-rank and sparse recovery framework for foreground detection.

III. BACKGROUND MODELING VIA MOTION-ASSISTED MATRIX RESTORATION

A. Motivation

The RPCA-based methods decompose the observed matrix (constructed by shaping each frame into a vector, and put vectors corresponding to successive frames as columns in the matrix) into two components. The low-rank component corresponds to the stationary background, while the sparse component represents the moving objects. Generally, the RPCA model fits well the background and foreground characteristics when foreground objects move fast: the latent background should be the same for all the frames within a scene (hence low-rank) and the foreground scatters in the spatio-temporal volume of the video clip (hence sparse). However, this prior assumption can be violated when the foreground occupies a large portion of the scene densely. Figure 2 shows a video clip containing two cars. The right car stays motionless all the time, and hence belongs to the background. As the left car moves slowly (belonging to foreground), background pixels are occluded by the car in many frames. In the observed matrix, each row corresponds to one pixel to be recovered in the background image, and the elements in a row are pixels from the background or the foreground along the temporal direction. As shown in Fig. 2 (c), many rows are dominated by the intensities of the left car, and the foreground components in these rows are thus dense, which does not meet the sparse assumption. As a result, the foreground information would

leak into the recovered background component. Therefore, previous RPCA-based methods present smearing artifacts around regions with slow motions or even camouflage. To overcome this shortcoming, it is desirable to find a smart way to let the model be aware of slow motions of foreground objects, which motivates us to propose a motion-assisted matrix restoration (MAMR) model for background-foreground separation.

B. Framework

The key idea of our MAMR method is to assign to each pixel a likelihood that it belongs to the background based on the estimated motion at that pixel. The background is to be extracted from K frames of a surveillance video clip denoted by $\{i_k\}_{k=0}^{K-1}$ of size $M \times N$. For easy mathematical manipulation, let \mathbf{i}_k be the vector form of frame i_k with the size $MN \times 1$. Then, we represent the frame sequences with matrix $\mathbf{D} = [\mathbf{i}_0, \mathbf{i}_1, \dots, \mathbf{i}_{K-1}]$ of size $MN \times K$. The recovered background component and foreground component in \mathbf{D} are denoted by \mathbf{B} and \mathbf{F} respectively. The aim is to separate \mathbf{B} and \mathbf{F} from \mathbf{D} . Denote a matrix, named weighting matrix, by \mathbf{W} whose elements represent the confidence levels that corresponding pixels in \mathbf{D} belong to the background. We propose to solve the foreground-background separation problem by solving the following optimization formula:

$$\min_{\mathbf{B}, \mathbf{F}} \|\mathbf{B}\|_* + \lambda \|\mathbf{F}\|_1, \text{ subject to } \mathbf{W} \circ \mathbf{D} = \mathbf{W} \circ (\mathbf{B} + \mathbf{F}), \quad (1)$$

where $\|\cdot\|_*$ and $\|\cdot\|_1$ denote the nuclear norm (sum of singular values) and ℓ_1 norm of a matrix, respectively, and “ \circ ” denotes element-wise multiplication of two matrices. Like previous methods, it is reasonable to assume the background as motionless in most practical surveillance applications (otherwise a global motion should be compensated). Under this assumption, any area with motion should not be considered as a part of background. Therefore, the weighting matrix \mathbf{W} is constructed from motion information (see Sec. III-C). Model (1) extends the classic matrix recovery model by taking the reliability of observed data into consideration. By incorporating motion information, areas dominated by slowly-moving objects are suppressed while the background that appears in only a few frames has more chances to be recovered in the final results.

C. Weighting Matrix Construction From Motion Information

Usually, the optical flow is computed pair-wise between two consecutive frames. However, for practical video clips, moving objects may move slowly or even stay motionless across many frames, i.e. camouflage, which are difficult to detect by optical flow. As shown in Fig. 6(a) (top row in red rectangle), the bag put on the carton by the left man is a camouflage across many frames. The optical flow between two adjacent frames is not sufficient to determine whether it belongs to foreground or background, resulting in misclassification. To remedy this problem, for each frame, we find a proper reference frame (called *anchor frame*, not necessarily the adjacent one) that differs from the current frame even in regions containing slowly moving foreground objects or even camouflages. Then, we estimate motion information for each frame referring to its

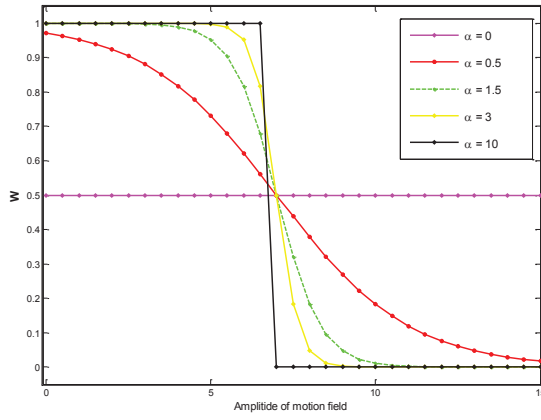


Fig. 3. The mapping from motion field ($\mathbf{o}_k^x, \mathbf{o}_k^y$) to the weighting matrix \mathbf{W} using sigmoid functions.

nearest anchor frame. Finally, we map the motion field into a weighting matrix.

1) *Dense Motion Estimation With Anchor Frame Selection*: For a single video, we set the first frame i_0 as the initial anchor frame. The remaining anchor frames are automatically selected according to the difference against the previous nearest anchor frame. To this end, the difference between the current frame i_k and the previous nearest anchor frame i_{anchor} is calculated for each frame. The difference e_k is defined as mean absolute difference (MAD) between two frames:

$$e_k = \frac{\sum_{m \in M, n \in N} |i_k^{m,n} - i_{anchor}^{m,n}|}{M \times N} \quad (2)$$

where m, n are the two-dimension pixel indexes in a frame. If the difference is larger than a threshold T , this frame is selected as a new anchor frame.

For each frame, we use the optical flow method in [37] to extract a dense motion field ($\mathbf{o}_k^x, \mathbf{o}_k^y$) between current video frame i_k and its previous nearest anchor frame, where \mathbf{o}_k^x and \mathbf{o}_k^y are the horizontal component and vertical component of the motion field, respectively. Both \mathbf{o}_k^x and \mathbf{o}_k^y are in the vector form in the same organization as i_k . Note that T should be chosen appropriately: too large a threshold would lead to few anchor frames, while too small a threshold would result in underestimation of motion (hence smearing artifacts around slowly-moving objects and misclassification of camouflages).

2) *Motion-to-Weight Mapping*: In the proposed model, the weighting matrix \mathbf{W} is constructed from the extracted dense motion field. We use the sigmoid function to map the motion field ($\mathbf{o}_k^x, \mathbf{o}_k^y$) into the weighting matrix. We define \mathbf{o}^x of size $MN \times K$ as the matrix form of horizontal motion fields for all frames in \mathbf{D} by stacking $\mathbf{o}_k^x, k = 0, 1, \dots, K-1$ as columns. Similarly, \mathbf{o}^y is defined for vertical motion fields. The weighting matrix \mathbf{W} is constructed as follows:

$$w_{jk} = 1 - \frac{1}{1 + \exp\left(\alpha \left(-\sqrt{\left(o_{jk}^x\right)^2 + \left(o_{jk}^y\right)^2} + \beta\right)\right)} \quad (3)$$

where α and β are the parameters of the sigmoid function which control the fitting slope and phase, respectively. β is

chosen according to the average intensity of the motion field. Note that α is a crucial parameter to shape the importance of motion information, as shown in Fig. 3: if α is zero, the weighting matrix \mathbf{W} is equal to 0.5 in all elements, and Model (1) turns into traditional RPCA-based method; As α increases, the slope of sigmoid function becomes steeper; when α takes very large values, e.g. 10, the sigmoid function will become approximately a step function, while \mathbf{W} also turns into a binary matrix, i.e., $\mathbf{W} \in \{0, 1\}^{MN \times K}$. Specifically, the weighting matrix \mathbf{W} is degraded from Formula (3) to the following binary mask:

$$w_{jk} = \begin{cases} 0, & \sqrt{\left(o_{jk}^x\right)^2 + \left(o_{jk}^y\right)^2} \geq \beta, \\ 1, & \text{otherwise.} \end{cases} \quad (4)$$

With such weighting, Model (1) becomes the following matrix completion model:

$$\min_{\mathbf{B}, \mathbf{F}} \|\mathbf{B}\|_* + \lambda \|\mathbf{F}\|_1, \text{ subject to } P_\Omega(\mathbf{D}) = P_\Omega(\mathbf{B} + \mathbf{F}), \quad (5)$$

where Ω denotes the linear subspace of entries in the observed matrix that belong to background for sure, and $P_\Omega(\cdot)$ is the associated projection operator.

D. The ADM-ALM Algorithm to Solve the MAMR Model

The MAMR model is essentially a convex optimization problem that can be solved by ADM-ALM method [38], [39]. The idea of ALM framework is to convert the original constrained optimization problem (1) to the minimization of the augmented Lagrangian function:

$$L(\mathbf{B}, \mathbf{F}, \mathbf{Y}, \mu) = \|\mathbf{B}\|_* + \lambda \|\mathbf{F}\|_1 + \langle \mathbf{Y}, \mathbf{W} \circ (\mathbf{D} - \mathbf{B} - \mathbf{F}) \rangle + \frac{\mu}{2} \|\mathbf{W} \circ (\mathbf{D} - \mathbf{B} - \mathbf{F})\|_F^2 \quad (6)$$

where μ is a positive constant, \mathbf{Y} is the Lagrangian multiplier. $\langle \cdot, \cdot \rangle$ denotes the matrix inner product, and $\|\cdot\|_F$ denotes the matrix Frobenius norm.

Instead of optimizing \mathbf{B} , \mathbf{F} and \mathbf{Y} simultaneously, the ADM solves \mathbf{B} , \mathbf{F} and \mathbf{Y} alternately:

$$\begin{cases} \mathbf{F}_{j+1} = \arg \min_{\mathbf{F}} \lambda \|\mathbf{F}\|_1 - \langle \mathbf{Y}_j, \mathbf{W} \circ \mathbf{F} \rangle + \frac{\mu_j}{2} \|\mathbf{W} \circ (\mathbf{D} - \mathbf{B}_j - \mathbf{F})\|_F^2, \\ \mathbf{B}_{j+1} = \arg \min_{\mathbf{B}} \|\mathbf{B}\|_* - \langle \mathbf{Y}_j, \mathbf{W} \circ \mathbf{B} \rangle + \frac{\mu_j}{2} \|\mathbf{W} \circ (\mathbf{D} - \mathbf{B} - \mathbf{F}_{j+1})\|_F^2, \\ \mathbf{Y}_{j+1} = \mathbf{Y}_j + \mu_j \mathbf{W} \circ (\mathbf{D} - \mathbf{B}_{j+1} - \mathbf{F}_{j+1}), \\ \mu_{j+1} = \rho \mu_j, \end{cases} \quad (7)$$

The solution of \mathbf{F}_{j+1} has the following closed-form

$$\mathbf{F}_{j+1} = \text{shrink}\left(\frac{1}{\mu_j} \mathbf{Y}_j + \mathbf{W} \circ (\mathbf{D} - \mathbf{B}_j), \frac{\lambda}{\mu_j}\right) \quad (8)$$

where $\text{shrink}(\cdot, \cdot)$ is the soft-thresholding function defined as:

$$\text{shrink}(\mathbf{X}, t) = \text{sign}(\mathbf{X}) \max(\text{abs}(\mathbf{X}) - t, 0) \quad (9)$$

The soft-thresholding operator applies on the matrix \mathbf{X} in an element-wise manner.

The solution of \mathbf{B}_{j+1} in (7) does not have a closed-form solution, and we resort to the accelerated proximal gradient algorithm [40] given as:

$$\begin{cases} (\mathbf{U}_l, \mathbf{S}_l, \mathbf{V}_l) = \\ \text{svd} \left(\frac{1}{\mu_j} \mathbf{Y}_j + \mathbf{W} \circ (\mathbf{D} - \mathbf{Z}_l) - \mathbf{F}_{j+1} + \mathbf{Z}_l \right), \\ \mathbf{B}_{l+1} = \mathbf{U}_l \text{shrink} \left(\mathbf{S}_l, \frac{1}{\mu_k} \right) \mathbf{V}_l^T, \\ \mathbf{Z}_{l+1} = \mathbf{B}_{l+1} + \frac{t_{l-1}-1}{t_l} (\mathbf{B}_{l+1} - \mathbf{B}_l), \\ t_{l+1} = 0.5(1 + \sqrt{1 + 4t_l^2}), \end{cases} \quad (10)$$

where t_l is a positive sequence with $t_1 = 1$, $\text{svd}(\cdot)$ denotes the singular value decomposition of a matrix.

The entire algorithm to solve problem (1) is summarized as Algorithm 1. In the ADM-ALM framework, the sub-problems are not necessarily solved exactly as long as the approximated solutions reduce the cost of Lagrangian function, which is therefore called inexact ALM [41]. Allowing inexact approximation of the sub-problems actually reduces overall computational complexity as the inner-loop iterations require considerable amount of computation to reach convergence. In our implementation, the inner loop for solving \mathbf{B}_{j+1} has only one iteration for acceleration.

The solution of Model (1), denoted by $(\mathbf{B}^*, \mathbf{F}^*)$, is obtained after the convergence of the iterative procedure: \mathbf{B}^* contains a background component for each frame, while \mathbf{F}^* provides a foreground component for each frame. We take the average of all columns in \mathbf{B}^* as the final recovered background image \bar{b} . Note that the ℓ_1 regularizer essentially describes signals that conform the Laplacian distribution. As a result, \mathbf{F}^* contains not only the desired foreground components but also noise leaked from background areas (due to the low-rank regularization). Therefore, we do not use \mathbf{F}^* as the foreground solution. Rather we extract foreground using the background subtraction approach with the recovered background \bar{b} (detailed in Sec. III-E).

Algorithm 1 ADM-ALM algorithm for the MAMR model.

Input: $\mathbf{D} \in \mathbb{R}^{MN \times K}$, $\mathbf{W} \in \mathbb{R}^{MN \times K}$, $\lambda > 0$, $\rho > 0$, $\mu > 0$;

Initialize: $\mathbf{F}_1 = 0, \mathbf{B}_1 = 0, \mathbf{Y}_1 = 0$;

while not converged do

$$\mathbf{F}_{j+1} = \text{shrink} \left(\frac{1}{\mu_j} \mathbf{Y}_j + \mathbf{W} \circ (\mathbf{D} - \mathbf{B}_j), \frac{\lambda}{\mu_j} \right);$$

$$t_1 = 1, \mathbf{Z}_1 = \mathbf{B}_j, \mathbf{B}_{j,1} = \mathbf{B}_j;$$

while not converged do

$$\begin{aligned} (\mathbf{U}_l, \mathbf{S}_l, \mathbf{V}_l) = \\ \text{svd} \left(\frac{1}{\mu_j} \mathbf{Y}_j + \mathbf{W} \circ (\mathbf{D} - \mathbf{Z}_l) - \mathbf{F}_{j+1} + \mathbf{Z}_l \right); \end{aligned}$$

$$\mathbf{B}_{j,l+1} = \mathbf{U}_l \text{shrink} \left(\mathbf{S}_l, \frac{1}{\mu_j} \right) \mathbf{V}_l^T;$$

$$\mathbf{Z}_{l+1} = \mathbf{B}_{j,l+1} + \frac{t_{l-1}-1}{t_l} (\mathbf{B}_{j,l+1} - \mathbf{B}_{j,l});$$

$$t_{l+1} = 0.5(1 + \sqrt{1 + 4t_l^2}), l = l + 1;$$

end while

$$\mathbf{B}_{j+1} = \mathbf{B}_{j,l+1};$$

$$\mathbf{Y}_{j+1} = \mathbf{Y}_j + \mu_j \mathbf{W} \circ (\mathbf{D} - \mathbf{B}_{j+1} - \mathbf{F}_{j+1});$$

$$\mu_{j+1} = \rho \mu_j, j = j + 1;$$

end while

Output: $(\mathbf{B}_j, \mathbf{F}_j)$;

E. Foreground Separation with Background Subtraction

Denote by \bar{f}_k the foreground image for frame i_k . The intensity value of \bar{f}_k at pixel x , denoted by $\bar{f}_k(x)$, is determined as:

$$\bar{f}_k(x) = \begin{cases} i_k(x) & \frac{\sum_{x \in \mathcal{N}_x} |i_k(x) - \bar{b}(x)|}{|\mathcal{N}_x|} > \tau + \sigma \\ 0 & \text{otherwise} \end{cases}, \quad (11)$$

where \mathcal{N}_x is the neighborhood of size $\omega \times \omega$ around x . $|\mathcal{N}_x|$ is the number of pixels in \mathcal{N}_x ; σ represents the level of noise variations in i_k ; τ is defined as:

$$\tau = \frac{\sum_{x \in \Phi} |i_k(x) - \bar{b}(x)|}{|\Phi|}, \quad (12)$$

where Φ is the set of pixels which contains non-zero values in $|i_k(x) - \bar{b}(x)|$; $|\Phi|$ is the number of non-zero pixels in the set Φ . By thresholding the average background subtraction image value over a small window, the outliers can be removed while the true foreground pixels are retained. For comparison in the experimental section, we convert the foreground image \bar{f}_k into a binary map by replacing the non-zero values in \bar{f}_k with 255.

F. Robust MAMR

In real applications, noise is quite ubiquitous. Usually, the data matrix is seriously damaged in some elements, while all of the elements would receive some lightweight noise pollution. Though the ℓ_1 norm can separate the intensive sparse errors from the intrinsic low-rank data matrix, it cannot deal with dense noise distributed over the whole frames. So, we propose a robust MAMR (RMAMR) model. We use the Frobenius norm to model dense noise. Denote by \mathbf{G} the error matrix of dense noise, the model can be formulated as follows:

$$\min_{\mathbf{B}, \mathbf{F}, \mathbf{G}} \|\mathbf{B}\|_* + \lambda \|\mathbf{F}\|_1 + \gamma \|\mathbf{G}\|_F^2, \quad (13)$$

$$\text{subject to } \mathbf{W} \circ \mathbf{D} = \mathbf{W} \circ (\mathbf{B} + \mathbf{F} + \mathbf{G}),$$

where γ is a positive constant, and $\|\cdot\|_F$ denotes the matrix Frobenius norm. The augmented Lagrangian function of problem (13) is given by

$$\begin{aligned} L(\mathbf{B}, \mathbf{F}, \mathbf{G}, \mathbf{Y}, \mu) = & \|\mathbf{B}\|_* + \lambda \|\mathbf{F}\|_1 + \gamma \|\mathbf{G}\|_F^2 \\ & + \langle \mathbf{Y}, \mathbf{W} \circ (\mathbf{D} - \mathbf{B} - \mathbf{F} - \mathbf{G}) \rangle \\ & + \frac{\mu}{2} \|\mathbf{W} \circ (\mathbf{D} - \mathbf{B} - \mathbf{F} - \mathbf{G})\|_F^2 \end{aligned} \quad (14)$$

Note that the difference between Model (1) and Model (13) is the introduction of the quadric term of \mathbf{G} . The solutions of \mathbf{B} and \mathbf{F} subproblems are similar to those in Model (1). So, we only present the solution of \mathbf{G} -subproblem:

$$\begin{aligned} \mathbf{G}_{j+1} = \arg \min_{\mathbf{G}} & \gamma \|\mathbf{G}\|_F^2 - \langle \mathbf{Y}_j, \mathbf{W} \circ \mathbf{G} \rangle \\ & + \frac{\mu_j}{2} \|\mathbf{W} \circ (\mathbf{D} - \mathbf{B}_j - \mathbf{F}_{j+1} - \mathbf{G})\|_F^2 \end{aligned} \quad (15)$$

The solution of \mathbf{G} has the following closed-form:

$$\mathbf{G}_{j+1} = \frac{1}{\mu_j + 2\gamma} (\mathbf{Y}_j + \mu_j \mathbf{W} \circ (\mathbf{D} - \mathbf{B}_j - \mathbf{F}_{j+1})) \quad (16)$$

The entire algorithm to solve problem (13) is summarized as Algorithm 2.

Algorithm 2 ADM-ALM algorithm for the RMAMR model.

Input: $\mathbf{D} \in \mathbb{R}^{MN \times K}$, $\mathbf{W} \in \mathbb{R}^{MN \times K}$, $\lambda > 0$, $\rho > 0$, $\gamma > 0$, $\mu > 0$;

Initialize: $\mathbf{F}_1 = 0$, $\mathbf{G}_1 = 0$, $\mathbf{B}_1 = 0$, $\mathbf{Y}_1 = 0$;

while not converged do

$$\mathbf{F}_{j+1} = \text{shrink} \left(\frac{1}{\mu_j} \mathbf{Y}_j + \mathbf{W} \circ (\mathbf{D} - \mathbf{B}_j - \mathbf{G}_j), \frac{\lambda}{\mu_j} \right);$$

$$\mathbf{G}_{j+1} = \frac{1}{\mu_j + 2\gamma} (\mathbf{Y}_j + \mu_j \mathbf{W} \circ (\mathbf{D} - \mathbf{B}_j - \mathbf{F}_{j+1}));$$

$$t_1 = 1, \mathbf{Z}_1 = \mathbf{B}_j, \mathbf{B}_{j,1} = \mathbf{B}_j;$$

while not converged do

$$(\mathbf{U}_l, \mathbf{S}_l, \mathbf{V}_l) =$$

$$\text{svd} \left(\frac{1}{\mu_j} \mathbf{Y}_j + \mathbf{W} \circ (\mathbf{D} - \mathbf{Z}_l) - \mathbf{F}_{j+1} - \mathbf{G}_{j+1} + \mathbf{Z}_l \right);$$

$$\mathbf{B}_{j,l+1} = \mathbf{U}_l \text{shrink} \left(\mathbf{S}_l, \frac{1}{\mu_j} \right) \mathbf{V}_l^T;$$

$$\mathbf{Z}_{l+1} = \mathbf{B}_{j,l+1} + \frac{t_{l-1}-1}{t_l} (\mathbf{B}_{j,l+1} - \mathbf{B}_{j,l});$$

$$t_{l+1} = 0.5(1 + \sqrt{1 + 4t_l^2}), l = l + 1;$$

end while

$$\mathbf{B}_{j+1} = \mathbf{B}_{j,l+1};$$

$$\mathbf{Y}_{j+1} = \mathbf{Y}_j + \mu_j \mathbf{W} \circ (\mathbf{D} - \mathbf{B}_{j+1} - \mathbf{F}_{j+1} - \mathbf{G}_{j+1});$$

$$\mu_{j+1} = \rho \mu_j, j = j + 1;$$

end while

Output: $(\mathbf{B}_j, \mathbf{F}_j, \mathbf{G}_j)$;

IV. EXPERIMENTAL RESULTS

In this section, we first present the setting of parameters in our algorithm (Sec. IV-A), and introduce test video clips and performance metrics used in our paper (Sec. IV-B). Then we investigate the parameters in weighting matrix construction that affect the recovery performance (Sec. IV-C), and compare different combining options to evaluate the impact of each module in our model (Sec. IV-D). Next, we compare our MAMR model with other state-of-the-art methods on challenging datasets in terms of background extraction (Sec. IV-E) and foreground detection (Sec. IV-F). In addition, we show the robustness to noise of our RMAMR model in Sec. IV-G. The running time is reported in Sec. IV-H.

In this paper, our method is compared with thirteen (13) methods: visual background extractor (ViBe) [21], self-organizing background subtraction (SOBS) [25], Gaussian mixture model (GMM) [13], statistical Bayesian segmentation and tracking (SBST) [23], pixel-based adaptive segmenter (PBAS) [22], fuzzy background modeling method (FBM) [28], Gaussian mixture model of Laurence Bender (LBG) [42], multi-layer background subtraction (MBS) [24], principal component pursuit (PCP) [15], outlier Pursuit (OP) [34], semi-soft GoDec algorithm (SSGoDec) [35], sparse Bayesian for low-rank matrix estimation (SBL) [36], and DEtecting Con-tiguous Outliers in the Low-rank Representation (DECOLOR) [9]. The codes for PCP, OP, SSGoDec, SBL are available at the project website [33], [43]. The codes for ViBe, GMM, SOBS, and DECOLOR are provided by the authors. The remaining methods are publicly available from Bgslibrary [44]. Since GMM, SOBS, LBG, MBS, and the RPCA-based

methods can generate the both background image and the binary foreground map, we compare the extracted backgrounds with these methods in (Sec. IV-E and Sec. IV-G). For all above algorithms, we seek optimal parameters around initial parameters published by the authors for fair comparison.

All the results are available in the project website¹. We direct interested readers to the website for more visual comparison results.

A. Parameter Setting

The parameters in our method fall into two categories: parameters (ρ, μ) that affect algorithm convergence and parameters $(T, \alpha, \beta, \lambda, \gamma, \sigma, \text{ and } \omega)$ that influence the performance.

1) *Convergence parameters:* μ is increasing during iterations from a small initial value $1/LSV(\mathbf{F})$, where $LSV(\cdot)$ takes the largest singular value of the operand matrix [38], [39]. In terms of ρ , too large a value would lead to unsatisfactory result, while too small one would slow the convergence rate of the algorithm. So we empirically set $\rho = 2$ for all the datasets.

2) *Performance parameters:* σ and ω are related to foreground detection. Thresholding factor σ in (11) depends on the level of noise and the average color difference between foreground pixels and background pixels in a video clip. It is chosen between the range [15, 35] for all the video clips (see Sec. IV-B). The neighborhood size ω in (11) is fixed at 3×3 .

The parameters T, α, β control the construction of weighting matrix. For each frame, T is adaptively set according to the average motion intensity over the previous processed K frames: $T = 1.3 \sum_{n=k-1}^{k-K} e_n / K$. If e_k is larger than T , the current frame is selected as a new anchor frame. β controls the turning point of the sigmoid function, and reflects the motion level beyond which is considered significant. In our implementation, β is chosen as the average intensity of the motion field, which is satisfactory for various datasets. Usually, α is set at a large value for a binary weighting matrix. The detailed discussion of α is given in Sec. IV-C.

The parameters λ and γ adjust the importance between low-rank term, sparse term, and noise term. In the noise-free case, our MAMR model set $\lambda = 10$, a large value that emphasize the importance of sparse regularization. In noisy case, our RMAMR model set $\lambda = 1$ and $\gamma = 1$ for the tested noise level.

B. Test Datasets and Performance Metrics

For comprehensive evaluation, we test our method on ten (10) video clips from ChangeDetection dataset (CDnet) [45] [46], and other two typical video clips *Monitor* and *Train*. CDnet contains six video categories with four to six video clips in each category. We choose the whole video clips from the category Dynamic Background, including *Boats*, *Canoe*, *Fall*, *Fountain01*, *Fountain02*, and *Overpass*; and pick one representative from each of other four categories, i.e., *Office* from Baseline, *Winterdrive (Winter)* from Intermittent Object Motion, *Boulevard* from Camera Jitter, and *PeopleInShade (Shade)* from Shadow. We pick continuous 200 frames from each

¹http://cs.tju.edu.cn/faculty/likun/projects/bf_separation/index.htm

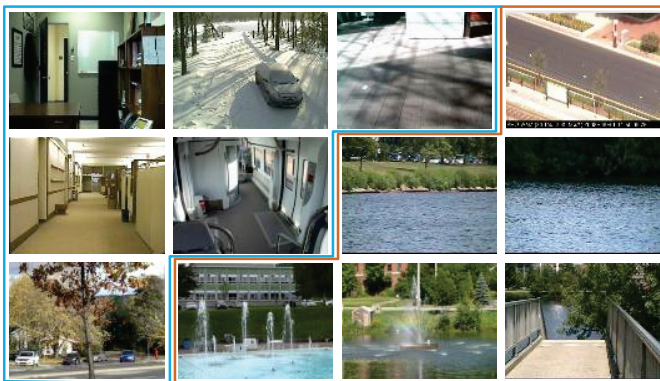


Fig. 4. Ground-truth background images for *Office*, *Winter*, *Shade*, *Monitor*, *Train*, *Fall* (enclosed by blue lines), and dynamic backgrounds excluding foregrounds for *Boulevard*, *Boats*, *Canoe*, *Fountain01*, *Fountain02*, *Overpass* (enclosed by yellow lines).

TABLE I

KEY INFORMATION OF THE TWELVE DATASETS INCLUDING TYPICAL CHARACTERISTIC APPEARING IN THEM. DIFFERENT DATASETS USED IN DIFFERENT EXPERIMENTAL SECTIONS ARE MARKED BY CHECK MARK.

Name	Characteristic	Sec.IVC	Sec.IVD	Sec.IVE	Sec.IVF	Sec.IVG
<i>Office</i>	Slowly-moving (man)	✓	✓	✓	✓	✓
<i>Winter</i>	Camouflage (right car), lighting variation	✓	✓	✓	✓	✓
<i>Boulevard</i>	Camera jitter, fast-moving (cars)			✓	✓	✓
<i>Shade</i>	Periodic motion, shadow (man)	✓		✓	✓	✓
<i>Monitor</i>	Periodic motion, slowly-moving	✓		✓	✓	✓
<i>Train</i>	Wagging (train), lighting variation	✓	✓	✓	✓	✓
<i>Boats</i>	Dynamic background (shimmering water)			✓	✓	✓
<i>Canoe</i>	Dynamic background (shimmering water)	✓	✓	✓	✓	✓
<i>Fall</i>	Dynamic background (waving tree)			✓	✓	✓
<i>Fountain01</i>	Dynamic background (spring)			✓	✓	✓
<i>Fountain02</i>	Dynamic background (spring)			✓	✓	✓
<i>Overpass</i>	Dynamic background (waving tree)	✓	✓	✓	✓	✓

dataset in the experiment. The key information of these twelve datasets is summarized in Table I. Each of these datasets may include various kinds of motions, lighting variations, camera jitter, camouflages, shadows, dynamic backgrounds, or the combination of them.

For objective evaluation in background extraction, ground-truth background images for static videos are created by averaging the background frames (without foreground included), which are manually picked from the sequence (shown in Fig. 4). We use the peak signal-to-noise ratio (PSNR) to measure the quality of extracted backgrounds against their ground-truth. Datasets with dynamic backgrounds are difficult to acquire their ground-truth backgrounds, and thus excluded in objective evaluation.

Foreground detection is essentially a binary segmentation task to classify each pixel into the background or foreground. We measure the objective performance of different algorithms by three metrics, namely *Recall* (*Re*), *Precision* (*Pre*), and *F*-

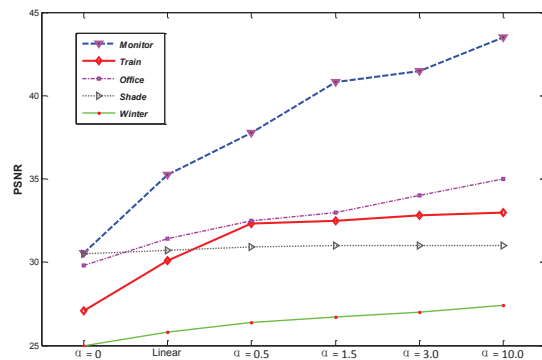


Fig. 5. Objective comparison with different values of α for recovered backgrounds on five video clips (static backgrounds). The values are computed against ground-truths in PSNR.

measure (F_1):

$$\begin{cases} Recall = tp/(tp + fn) \\ Precision = tp/(tp + fp) \\ F_1 = (2 \times Recall \times Precision)/(Recall + Precision) \end{cases} \quad (17)$$

where tp (true positive) represents correctly classified foreground pixels, fn (false negative) denotes the number of foreground pixels incorrectly classified as background, fp (false positive) stands for the total number of background pixels incorrectly classified as foreground. *Precision* gives the percentage of correctly detected foreground pixels among all detected foreground pixels. *Recall* weighs the percentage of correctly detected foreground pixels among the total number of foreground pixels. *F-measure* is the weighted harmonic mean of *Precision* and *Recall*, which measures the overall detection quality of an algorithm. For all the three metrics, the higher the value is, the better the performance it has.

C. Effect of α in Motion-to-Weight Mapping

Note that α is a crucial parameter to map the motion field ($\mathbf{o}_k^x, \mathbf{o}_k^y$) into weighting matrix \mathbf{W} . We sample five values of α , i.e., 0, 0.5, 1.5, 3, and 10 (which generates a nearly binary matrix) to investigate how α affect the recovery performance. A linear mapping is also tested between \mathbf{W} and ($\mathbf{o}_k^x, \mathbf{o}_k^y$). Fig. 5 shows our objective results on recovered backgrounds. As α increases, the recovered performance gets better for each video clip, and reaches the highest PSNR when α equals to 10 (approximately binary weight). This trend is particularly significant for *Monitor* and *Train*, because *Monitor* contains a slowly-walking men while the runaway thief occupies most space of the picture across many frames in *Train*.

Figure 6 further shows two datasets under different alpha parameters. The ghosting artifacts are eliminated as α increases, and the best performance is achieved when $\alpha = 10$. In Fig. 6, we observe that the bag on the carton (highlighted with a red rectangle) is successfully removed from the background when $\alpha > 1.5$. The results show that our method favors the binary weights to have the most accurate separation result. So we use Model (5) for our method in the following results.

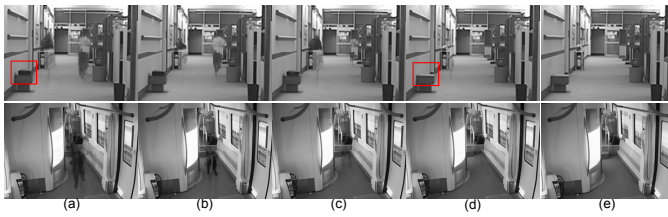


Fig. 6. Visual quality comparison for recovered background in *Monitor* (the top row) and *Train* (the bottom row) using the proposed method under different α values. From left to right (a) PCP ($\alpha = 0$) [15], (b) $\alpha = 0.5$, (c) $\alpha = 1.5$, and (d) $\alpha = 3$, (e) $\alpha = 10$.

TABLE II
QUANTITATIVE FOREGROUND DETECTION RESULTS ON DIFFERENT COMBINING OPTIONS. OPT1-OPT10 REPRESENT TEN COMBINING OPTIONS, IN WHICH OF1-OF4 ARE FOUR OPTICAL FLOWS, AND AFS DENOTES THE ANCHOR FRAMES SELECTION.

	Opt1	Opt2	Opt3	Opt4	Opt5	Opt6	Opt7	Opt8	Opt9	Opt10
Recall	0.65	0.73	0.79	0.79	0.81	0.80	0.83	0.84	0.83	0.85
Pre	0.72	0.74	0.73	0.73	0.75	0.75	0.79	0.78	0.87	0.86
F1	0.67	0.71	0.76	0.78	0.78	0.78	0.81	0.81	0.83	0.85

Opt1	Opt2	Opt3	Opt4	Opt5
OF4 + GMM	OF4 + AFS + GMM	OF1 + RPCA	OF2 + RPCA	OF3 + RPCA

Opt6	Opt7	Opt8	Opt9	Opt10
OF4 + RPCA	OF1 + AFS + RPCA	OF2 + AFS + RPCA	OF3 + AFS + RPCA	OF4 + AFS + RPCA

D. Performance of Our Method with Different Combining Options

In this section, we test different combining options to verify the importance of different modules included in our MAMR model. Four optical flow computation methods, i.e., OF1 by Black and Anandan [47], OF2 by Liu *et al.* [37], OF3 by Sun *et al.* [48], OF4 by Brox *et al.* [49], are used to derive the weight matrix. These methods provide different tradeoffs between speeds and accuracy. In total, ten different combining options are designed for comprehensive comparison (shown in Table II). The acronyms of the combining options are also explained in Table II. For Opt1 and Opt2, the compound data of optical flows and pixels values are modeled with GMM, in which the background image is updated on-the-fly, and foreground is detected by comparing its probability belonging to the foreground over that belonging to the background. The quantitative results and visual comparison, are given in Table II and Fig. 7, respectively.

As shown in Table II, different optical flows obtain almost the same results under the same type of combinations (OF + RPCA or OF + AFS + RPCA). Therefore, we choose the fastest optical flow algorithm (OF2) [37] to accelerate our method. Comparing the results of OF+RPCA and OF+AFS+RPCA, we observe that the performance would decline if the AFS is excluded, which demonstrates the effectiveness of AFS to detect moving objects.

Observing the effect of the weighting matrix constructed from optical flow with anchor frames selection, one may want to see the effect of using this weighting matrix with other

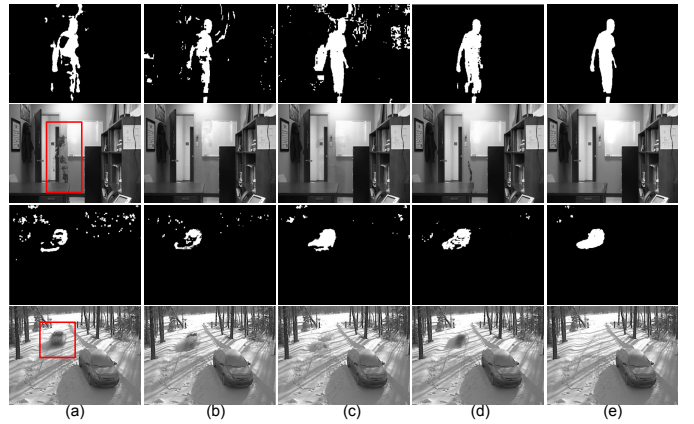


Fig. 7. Binary foreground maps and its corresponding extracted backgrounds obtained with different combining options on the 656th frame of *Office* and the 1936th frame of *Winter*: (a) GMM [13], (b) Opt1 (OF4 [49] + GMM), (c) Opt2 (OF4 [49] + AFS + GMM), (d) Opt6 (OF4 [49] + RPCA), (e) Opt10 (OF4 [49] + AFS + RPCA). AFS is the short for anchor frames selection.

models such as GMM. To this end, we replace the RPCA model with the GMM model in OPT1 and OPT2. As shown in Table II, the replacement of RPCA with GMM suffers from severe performance loss. In Fig. 7, we show the performance evolution in a more intuitive way with visual comparison. Only using GMM on color information cannot estimate the foreground precisely, e.g., the man in *Office* and the car in *Winter*. When adding motion information (Opt1), the results are improved, but some regions still cannot be detected due to the failure of frame-by-frame optical flow computation in detecting slowly-moving objects. By further introducing anchor frame selection (Opt2), most pixels of the foreground can be found. However, there are still some smearing artifacts due to the background variations. The results generated by our method (Opt10), shown in Fig. 7(e), are more accurate, and the recovered backgrounds are more close to the groundtruth. Experimental results in this section verify that each module of our method plays an important role in improving the performance, and the assembling of the three components in our method show great power towards accurate background-foreground separation.

E. Experimental Results on Background Extraction

Figure 8 compares backgrounds extracted by SOBS [25], LBG [42], MBS [24], PCP [15], DECOLOR [9], and our MAMR. We test all the video clips, but present the results for only the most challenging seven ones to save space (see the project website for the results on all the video clips). For the same reason, of the five RPCA-based methods, we present the results for only the baseline PCP [15] and the most recent DEOLOR [9]. The results in Fig. 8 show that our method provides significant improvement over other methods. The background images recovered by our MAMR model are more close to groundtruth while the ones extracted by other methods present smearing and ghosting artifacts.

For *Boulevard*, *Fall*, *Fountain01*, and *Fountain02*, the foreground objects are small and run fast in the scenes. For this type of motions, all the methods can recover promis-

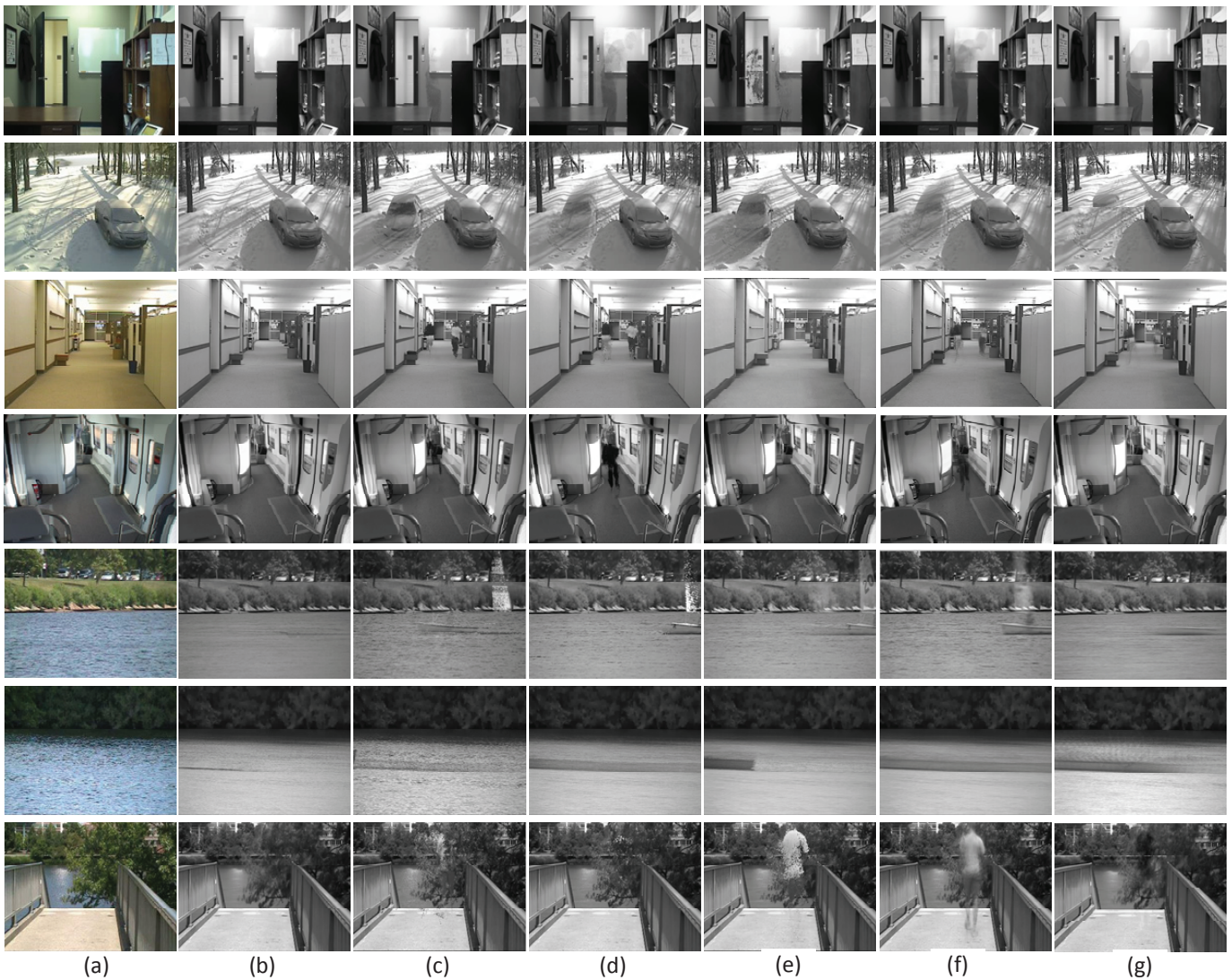


Fig. 8. Visual quality comparison for background extraction on seven video clips. (a) true backgrounds, (b) MAMR, (c) SOBS [25], (d) LBG [42], (e) MBS [24], (f) PCP [15], and (g) DEOLOR [9]. From top to bottom present extracted backgrounds for *office*, *Winter*, *Monitor*, *Train*, *Boats*, *Canoe*, and *Overpass*, respectively.

ing background images. However, when it comes to slowly-moving objects, e.g., the walking men in *Office*, *Overpass*, and *Monitor*, and the running boats in *Boats* and *Canoe*, results produced by the compared methods present severe smearing artifacts. This is because the slowly-moving objects occlude the scene across many frames, which may be considered as a part of background, resulting in the failure of background extraction. Moreover, for *Winter*, the left car keeps motionless at first, and moves very slowly during the whole video (nearly camouflage). SOBS, LBG, and MBS tend to classify the intermittent moving object as background and fail to adapt to background changes. The RPCA-based methods, i.e., PCP and DECOLOR, present smearing artifacts along the trajectories of running car. On the contrary, our method achieves promising results for all the evaluation datasets. With the help of motion information, we can prevent the slow moving objects (e.g., motionless man, running boat) from leaking into backgrounds, and recover the accurate backgrounds without smearing and

ghosting artifacts.

F. Experimental Results on Foreground Detection

With the extracted background, we detect foreground objects via background subtraction. Foreground detection results are reported in Table III. Our method achieves the highest *F-measure* for all the datasets, though some values in terms of the *Precision* and *Recall* metrics are a little lower than other methods. For FBM, SBL and DECOLOR, the values of *Precision* and *Recall* present a trend that if the value of one metric is very high, the other would be very low. For *Monitor*, SBST achieves the highest *Recall* (0.97), but extremely low *Precision* (only 0.42). As a result, these methods have low *F-measure* values. In contrast, our method obtains high values in terms of both *Precision* and *Recall*, and therefore has high *F-measure* values. This proves the superior performance of our method over other methods.

Figure 9 further presents visual comparison results of fore-

TABLE III
QUANTITATIVE FOREGROUND SEPARATION RESULTS IN TERMS OF RECALL, PRECISION, AND F-MEASURE ON THE TWELVE VIDEO CLIPS.

	<i>Office</i>			<i>Winter</i>			<i>Boulevard</i>			<i>Shade</i>			<i>Monitor</i>			<i>Train</i>			<i>Boats</i>			<i>Canoe</i>			<i>Fall</i>			<i>Fountain01</i>			<i>Fountain02</i>			<i>Overpass</i>		
	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁
MAMR	0.85	0.86	0.85	0.68	0.72	0.70	0.68	0.75	0.73	0.80	0.81	0.80	0.94	0.94	0.94	0.76	0.93	0.84	0.87	0.38	0.23	0.78	0.87	0.81	0.72	0.80	0.75	0.74	0.42	0.51	0.97	0.95	0.96	0.78	0.87	0.82
VIBe[21]	0.70	0.80	0.69	0.57	0.18	0.23	0.75	0.42	0.50	0.74	0.78	0.76	0.61	0.97	0.75	0.69	0.64	0.67	0.59	0.19	0.11	0.80	0.67	0.73	0.84	0.58	0.44	0.74	0.27	0.16	0.94	0.85	0.78	0.66	0.78	0.64
SOBS[25]	0.67	0.79	0.69	0.18	0.53	0.24	0.37	0.61	0.43	0.63	0.78	0.68	0.77	0.93	0.84	0.90	0.46	0.61	0.27	0.11	0.07	0.54	0.70	0.64	0.74	0.62	0.61	0.30	0.61	0.32	0.90	0.82	0.88	0.64	0.78	0.68
GMM[13]	0.53	0.82	0.59	0.39	0.58	0.45	0.57	0.56	0.46	0.75	0.71	0.72	0.74	0.78	0.76	0.57	0.15	0.24	0.29	0.21	0.16	0.51	0.62	0.49	0.97	0.60	0.43	0.84	0.36	0.22	0.94	0.95	0.95	0.64	0.78	0.74
SBST[23]	0.78	0.56	0.61	0.81	0.45	0.57	0.87	0.11	0.19	0.73	0.47	0.54	0.97	0.42	0.58	0.86	0.12	0.21	0.59	0.16	0.09	0.82	0.11	0.19	0.99	0.18	0.10	0.30	0.02	0.05	0.72	0.55	0.38	0.27	0.39	0.24
PBAS[22]	0.78	0.76	0.77	0.47	0.39	0.38	0.65	0.75	0.58	0.73	0.77	0.75	0.91	0.81	0.86	0.73	0.62	0.67	0.27	0.11	0.07	0.62	0.94	0.74	0.99	0.79	0.65	0.67	0.49	0.34	0.97	0.92	0.88	0.72	0.84	0.75
FBM[28]	0.62	0.76	0.71	0.37	0.36	0.34	0.50	0.73	0.52	0.65	0.77	0.70	0.61	0.99	0.75	0.60	0.63	0.61	0.26	0.16	0.11	0.35	0.75	0.45	0.75	0.65	0.58	0.37	0.29	0.24	0.95	0.91	0.96	0.80	0.62	0.69
PCP[15]	0.57	0.76	0.62	0.55	0.40	0.42	0.51	0.80	0.62	0.69	0.74	0.71	0.80	0.90	0.84	0.70	0.85	0.74	0.09	0.05	0.06	0.36	0.32	0.33	0.59	0.47	0.50	0.56	0.06	0.11	0.75	0.77	0.75	0.52	0.87	0.63
OP[34]	0.52	0.60	0.54	0.50	0.50	0.58	0.47	0.54	0.58	0.60	0.50	0.55	0.70	0.69	0.70	0.68	0.83	0.71	0.21	0.06	0.10	0.48	0.33	0.39	0.61	0.19	0.28	0.30	0.02	0.02	0.45	0.37	0.42	0.61	0.80	0.68
SSGoDec[35]	0.67	0.73	0.66	0.61	0.52	0.55	0.62	0.57	0.60	0.72	0.74	0.73	0.77	0.89	0.78	0.73	0.70	0.71	0.34	0.13	0.18	0.64	0.35	0.42	0.64	0.37	0.45	0.60	0.06	0.11	0.72	0.83	0.77	0.66	0.73	0.66
SBL[36]	0.58	0.74	0.61	0.68	0.56	0.62	0.73	0.80	0.73	0.71	0.74	0.72	0.90	0.79	0.73	0.68	0.85	0.74	0.16	0.08	0.10	0.34	0.44	0.35	0.51	0.53	0.50	0.56	0.07	0.12	0.80	0.72	0.74	0.60	0.79	0.64
DECOLOR[9]	0.87	0.61	0.71	0.64	0.70	0.69	0.86	0.70	0.72	0.73	0.32	0.42	0.85	0.64	0.79	0.72	0.83	0.77	0.36	0.12	0.19	0.70	0.85	0.73	0.76	0.53	0.61	0.88	0.04	0.08	0.97	0.61	0.74	0.82	0.72	0.81

ground detection for one typical frame in each video clip. We only choose some typical methods to show the results to save space. For slowly-moving objects in *Office*, *Monitor*, and camouflage in *Winter*, the proposed method accurately detects the foreground objects. For *Winter*, *Shade* and *Train*, due to the poor lighting conditions and shadows cast by foreground objects, most methods fail to detect the intact foreground. It is more tough to handle backgrounds with varying ambient lighting variations and shadows than static ones since these can cause fake motions in the background. For *Shade*, *Monitor*, and *Overpass*, they contain nearly periodic motions as the man in each scene repeats the action of walking and the poses are similar across frames. Our method successfully detects this type of motions and recover accurate foregrounds; DECOLOR [9] also provides similar results.

The most difficult category on detecting foreground is the Dynamic Background. Due to the motions in the background, such as the running water in *Boats*, *Canoe*, the waving trees in *Fall*, *Overpass*, and springs in *Fountain01*, *Fountain02*, the judge on whether the pixel belongs to foreground or background is very difficult. For example, in *Boats*, all the methods fail to detect the body of the boat while our MAMR model is able to faithfully separate the boat; in *Fall*, most methods cannot fend against the influence of the waving tree, and the foreground masks are polluted severely. DECOLOR provides comparable results to our methods and ensures the integrity of the foreground, but also yields overestimation in some cases, which can be observed from the man in both *Shade* and *Overpass*. Moreover, for *Fountain01*, the flowing fountain water is misclassified as part of foreground and further expanded due to the smoothness regularization.

In general, our method significantly outperforms other methods. The results of our MAMR model are the closest to ground truth binary maps. Through encoding motion cues into RPCA, our motion-aware method significantly improves the performance of other motion-unaware RPCA methods.

TABLE V
QUANTITATIVE BACKGROUND EXTRACTION RESULTS IN TERMS OF PSNR ON TWELVE NOISY VIDEO CLIPS.

	<i>Office</i>	<i>Winter</i>	<i>Shade</i>	<i>Monitor</i>	<i>Train</i>
RMAMR	34.61	27.12	31.20	39.87	33.06
SOBS[25]	25.02	20.60	27.16	32.26	23.38
LBG[42]	33.06	26.33	30.93	37.65	29.44
MBS[24]	27.56	25.12	28.74	34.99	27.71
PCP[15]	24.16	21.73	25.34	29.90	30.50
OP[34]	21.08	20.12	21.23	30.75	32.50
SSGoDec[35]	29.89	24.19	27.50	35.43	33.02
SBL[36]	29.72	25.00	28.61	36.32	33.04
DECOLOR[9]	29.83	26.02	31.02	37.20	33.04

G. Experimental Results on Noisy Datasets

In this section, we test the performance of our RMAMR model against noisy datasets. To this end, we add Gaussian noise with a variance of 25 to the original test clips. The noise degradation can affect a lot on background extraction and foreground detection.

Objective recovery results of foreground detection and background extraction are reported in Table IV and Table V, respectively. As shown in Table IV, though most methods including ours obtain a lower metric values than results on clean datasets (Table III in Sec. IV-F), our method still obtains the best objective values for most cases, which demonstrates robustness of our RMAMR model to noise. In Table V, our method obtains the highest PSNRs against the groundtruth backgrounds. Note that all the RPCA-based methods achieve satisfactory denoising results, which have relative higher values of PSNR than other methods. Figure 10 further presents visual comparisons of foreground detection results. Our method generates almost the same foreground results as those on clean datasets, while other methods tend to produce noisy results due to the presence of noise.



Fig. 9. Visual quality comparison for foreground detection on twelve video clips: (a) input image frame and (b) corresponding groundtruth binary foreground, (c) our MAMR model, (d) ViBe [21], (e)SOBS [25], (f)GMM [13], (g) FBM [28], (h) PCP [15], and (i) DECOLOR [9]. From top to bottom present the 656th frame of *Office*, the 1936th frame of *Winter*, the 816th frame of *Boulevard*, the 481th frame of *Shade*, the 56th frame of *Monitor*, the 46th frame of *Train*, the 7101th frame of *Boats*, the 956th frame of *Canoe*, the 1497th frame of *Fall*, the 717th frame of *Fountain01*, the 741th frame of *Fountain02*, and the 2401th frame of *Overpass*, respectively. The gray regions in the ground-truths provided by CDnet are excluded when making objective comparison.

H. Running Time

Our method mainly consists of two parts: dense motion estimation by optical flow [37] and convex programming in solving the MAMR/RMAMR models. We report running time for *Fountain01* with 40 frames of size 320×240 . The ADM-ALM algorithms are implemented in MATLAB (R2013a), and

run on a desktop with a 3.4 GHz Core4 i7 processor and 8 GB memory.

The motion estimation takes about 20 seconds on average to process 40 frames (each frame takes about 0.5 seconds). The ALM-ADM algorithm takes 2.53 seconds to separate the background and foreground from the 40-frame sequence by solving Model (5), while takes 2.60 seconds solving Model

TABLE IV
 QUANTITATIVE FOREGROUND SEPARATION RESULTS IN TERMS OF RECALL, PRECISION, AND F-MEASURE ON TWELVE NOISY VIDEO CLIPS.

	Office			Winter			Boulevard			Shade			Monitor			Train			Boats			Canoe			Fall			Fountain01			Fountain02			Overpass		
	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁	Re	Pre	F ₁
RMAMR	0.86	0.84	0.84	0.70	0.60	0.67	0.65	0.75	0.70	0.79	0.82	0.80	0.83	0.94	0.88	0.71	0.83	0.77	0.85	0.38	0.22	0.78	0.82	0.79	0.71	0.73	0.72	0.70	0.39	0.46	0.90	0.90	0.93	0.80	0.83	0.81
SOBS[25]	0.69	0.70	0.67	0.19	0.41	0.18	0.41	0.67	0.48	0.65	0.82	0.70	0.66	0.98	0.78	0.25	0.52	0.34	0.27	0.12	0.06	0.54	0.70	0.58	0.68	0.60	0.61	0.32	0.58	0.31	0.86	0.81	0.84	0.64	0.80	0.67
LBG[42]	0.67	0.84	0.78	0.64	0.37	0.42	0.63	0.46	0.44	0.74	0.89	0.79	0.78	0.93	0.85	0.77	0.40	0.53	0.34	0.11	0.13	0.68	0.56	0.54	0.62	0.48	0.53	0.60	0.06	0.13	0.76	0.80	0.78	0.66	0.82	0.71
MBS[24]	0.84	0.32	0.44	0.33	0.58	0.41	0.49	0.65	0.52	0.82	0.44	0.54	0.64	0.97	0.77	0.68	0.71	0.70	0.08	0.05	0.05	0.57	0.47	0.42	0.54	0.46	0.47	0.62	0.56	0.37	0.74	0.82	0.73	0.34	0.57	0.39
PCP[15]	0.52	0.80	0.60	0.44	0.40	0.41	0.50	0.72	0.60	0.68	0.73	0.69	0.70	0.86	0.77	0.76	0.66	0.71	0.09	0.05	0.06	0.35	0.30	0.32	0.58	0.46	0.50	0.52	0.06	0.09	0.83	0.62	0.70	0.63	0.77	0.67
OP[34]	0.32	0.58	0.43	0.30	0.42	0.37	0.45	0.35	0.40	0.58	0.47	0.53	0.68	0.69	0.68	0.68	0.82	0.71	0.22	0.06	0.10	0.20	0.32	0.26	0.61	0.19	0.28	0.27	0.02	0.02	0.52	0.36	0.31	0.71	0.56	0.58
SSGoDec[35]	0.62	0.74	0.63	0.60	0.51	0.53	0.60	0.53	0.55	0.71	0.72	0.71	0.73	0.82	0.81	0.69	0.70	0.70	0.31	0.12	0.17	0.62	0.37	0.42	0.53	0.46	0.48	0.58	0.06	0.10	0.79	0.69	0.73	0.59	0.84	0.67
SBL[36]	0.57	0.72	0.60	0.65	0.53	0.60	0.70	0.79	0.70	0.70	0.70	0.70	0.82	0.80	0.80	0.68	0.85	0.73	0.15	0.08	0.10	0.33	0.44	0.35	0.51	0.53	0.50	0.56	0.07	0.12	0.73	0.83	0.77	0.47	0.86	0.59
DECOLOR[9]	0.82	0.65	0.70	0.64	0.73	0.67	0.87	0.70	0.70	0.71	0.35	0.40	0.83	0.68	0.78	0.73	0.81	0.76	0.35	0.13	0.18	0.67	0.83	0.74	0.76	0.53	0.60	0.84	0.06	0.10	0.96	0.67	0.79	0.81	0.73	0.76

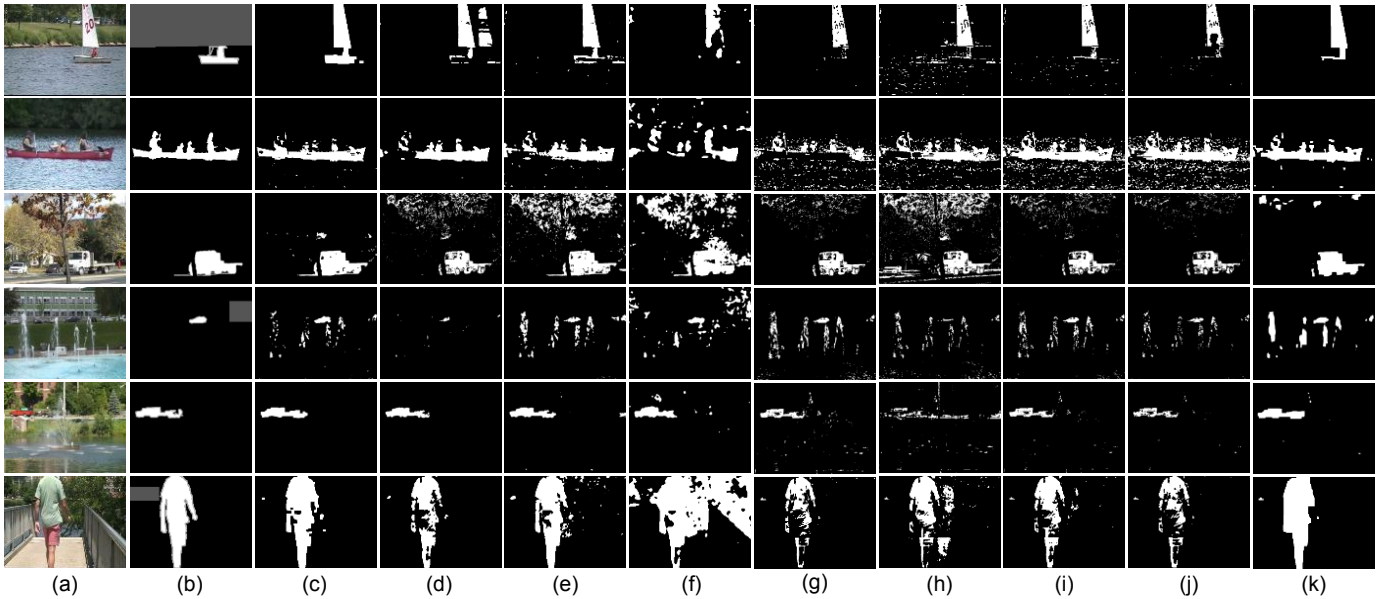


Fig. 10. Visual quality comparison for binary foreground maps on the synthetic noisy video clips on the category of Dynamic Background: (a) Original noisy frame, (b) Ground truth, (c) our RMAMR model, (d) SOBS [25], (e) LBG [42], (f) MBS [24], (g) PCP [15], (h) OP [34], (i) SSGoDec [35], (j) SBL [36], and (k) DECOLOR [9].

(13), which is comparable to the RPCA-based method (2.26 seconds) [15]. Besides, the optical flow method [37] can be replaced by other faster motion estimators.

V. CONCLUSION

In this paper, we propose a motion-assisted matrix restoration (MAMR) model for foreground-background separation from video clips. In the proposed MAMR model, the backgrounds across frames are modeled by a low-rank matrix, while the foreground objects are modeled by a sparse matrix. To facilitate efficient foreground-background separation, a dense motion field is estimated for each frame, and mapped into a weighting matrix to assign the likelihood of pixels belonging to the background. Anchor frames are selected in the dense motion estimation to overcome the difficulty of detecting slowly-moving objects and camouflages. We also extend our model to a robust MAMR model (RMAMR). Experimental results demonstrate our method is quite versatile

for surveillance videos with different types of motions and lighting conditions.

The proposed framework could be improved and extended in future work: 1) exploit the incorporation of more complex motion models or other clues into the low-rank and sparse recovery framework for foreground detection, 2) optimize model parameters according to video characteristics, 3) explore weighted versions of more low-rank and sparse recovery models as well as their applications to other image processing tasks.

ACKNOWLEDGEMENTS

J. Yang is the corresponding author for this paper. The authors are grateful to the reviewers' comments that improved the paper.

REFERENCES

- [1] H. Li, L. Yu, and W. Li, "Wireless scalable video coding using a hybrid digital-analog scheme," *IEEE TCSVT*, vol. 24, no. 2, pp. 331–345, 2013.

- [2] Y. Fang, W. Lin, Z. Chen, C. Tsai, and C. Lin, "A video saliency detection model in compressed domain," *IEEE TCSVT*, vol. 24, no. 1, pp. 27–38, 2013.
- [3] P. V. K. Borges, N. Conci, and A. Cavallaro, "Video-based human behavior understanding: a survey," *IEEE TCSVT*, vol. 23, no. 11, pp. 1993–2008, 2013.
- [4] J. K. Suhr, H. G. Jung, G. Li, and J. Kim, "Mixture of Gaussians-based background subtraction for bayer-pattern image sequences," *IEEE TCSVT*, vol. 21, no. 3, pp. 365–370, 2011.
- [5] C.-C. Chiu, M.-Y. Ku, and L.-W. Liang, "A robust object segmentation system using a probability-based background extraction algorithm," *IEEE TCSVT*, vol. 20, no. 4, pp. 518–528, 2010.
- [6] P.-M. Jodoin, M. Mignotte, and J. Konrad, "Statistical background subtraction using spatial cues," *IEEE TCSVT*, vol. 17, no. 12, pp. 1758–1763, 2007.
- [7] A. M. McIvor, "Background subtraction techniques," in *Proc. Image and Vision Computing*, vol. 1, no. 3, pp. 155–163, 2000.
- [8] S.-C. Huang, "An advanced motion detection algorithm with video quality analysis for video surveillance systems," *IEEE TCSVT*, vol. 21, no. 1, pp. 1–14, 2011.
- [9] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE TPAMI*, vol. 35, no. 3, pp. 597–610, 2013.
- [10] Y. Tsai and A. Averbuch, "Automatic segmentation of moving objects in video sequences: a region labeling approach," *IEEE TCSVT*, vol. 12, no. 7, pp. 597–612, 2002.
- [11] B. Dey and M. K. Kundu, "Robust background subtraction for network surveillance in H.264 streaming video," *IEEE TCSVT*, vol. 23, no. 10, pp. 1695–1703, 2013.
- [12] S. Brutzer, B. Hoferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Proc. CVPR*, 2011, pp. 1937–1944.
- [13] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. ICPR*, vol. 2, 2004, pp. 28–31.
- [14] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.
- [15] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.
- [16] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE TPAMI*, vol. 19, no. 7, pp. 780–785, 1997.
- [17] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE TPAMI*, vol. 25, no. 10, pp. 1337–1342, 2003.
- [18] A. El Maadi and X. Maldague, "Outdoor infrared video surveillance: A novel dynamic technique for the subtraction of a changing background of ir images," *Infrared physics & technology*, vol. 49, no. 3, pp. 261–265, 2007.
- [19] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-Based Surveillance Systems*, 2002, pp. 135–144.
- [20] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. CVPR*, vol. 2, 1999.
- [21] O. Barnich and M. Van Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," *IEEE TIP*, vol. 20, no. 6, pp. 1709–1724, 2011.
- [22] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Proc. IEEE CVPRW*, 2012, pp. 38–43.
- [23] A. B. Godbehere, A. Matsukawa, and K. Goldberg, "Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation," in *American Control Conference (ACC)*. IEEE, 2012, pp. 4305–4312.
- [24] J. Yao and J.-M. Odobez, "Multi-layer background subtraction based on color and texture," in *Proc. CVPR*, 2007, pp. 1–8.
- [25] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE TIP*, vol. 17, no. 7, pp. 1168–1177, 2008.
- [26] A. Manzanera, " Σ - Δ background subtraction and the zipf law," in *Progress in Pattern Recognition, Image Analysis and Applications*. Springer, 2007, pp. 42–51.
- [27] Y. Zhou, W. Xu, H. Tao, and Y. Gong, "Background segmentation using spatial-temporal multi-resolution MRF," in *Proc. IEEE Workshop on Application of Computer Vision (WACV)*, vol. 2, 2005, pp. 8–13.
- [28] Z. Zhao, T. Bouwmans, X. Zhang, and Y. Fang, "A fuzzy background modeling approach for motion detection in dynamic backgrounds," in *Multimedia and Signal Processing*, 2012, pp. 177–185.
- [29] M. Seki, T. Wada, H. Fujiwara, and K. Sumi, "Background subtraction based on cooccurrence of image variations," in *Proc. CVPR*, vol. 2, 2003, pp. II–65.
- [30] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE TPAMI*, vol. 22, no. 8, pp. 831–843, 2000.
- [31] Z. Gao, L.-F. Cheong, and M. Shan, "Block-sparse RPCA for consistent foreground detection," in *Proc. ECCV*. Springer, 2012, pp. 690–703.
- [32] C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Foreground detection based on low-rank and block-sparse matrix decomposition," in *Proc. IEEE ICIP*, 2012, pp. 1225–1228.
- [33] T. Bouwmans and E. H. Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *CVIU*, vol. 122, pp. 22–34, 2014.
- [34] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Advances in Neural Information Processing Systems*, 2010, pp. 2496–2504.
- [35] T. Zhou and D. Tao, "Godec: Randomized low-rank & sparse matrix decomposition in noisy case," in *Proc. ICML*, 2011, pp. 33–40.
- [36] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse Bayesian methods for low-rank matrix estimation," *IEEE TSP*, vol. 60, no. 8, pp. 3964–3977, 2012.
- [37] C. Liu, "Beyond pixels: exploring new representations and applications for motion analysis," Ph.D. dissertation, Citeseer, 2009.
- [38] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.
- [39] J. Yang and Y. Zhang, "Alternating direction algorithms for ℓ_1 -problems in compressive sensing," *SIAM journal on scientific computing*, vol. 33, no. 1, pp. 250–278, 2011.
- [40] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," *Pacific Journal of Optimization*, vol. 6, no. 615–640, p. 15, 2010.
- [41] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, vol. 61, 2009.
- [42] T. Bouwmans, F. El Baf, and B. Vachon, "Background modeling using mixture of Gaussians for foreground detection—a survey," *Recent Patents on Computer Science*, vol. 1, no. 3, pp. 219–237, 2008.
- [43] C. Guyon, T. Bouwmans, E.-h. Zahzah *et al.*, "Robust principal component analysis for background subtraction: Systematic evaluation and comparative analysis," *Principal Component Analysis, P. Sanguansat, Ed*, 2012.
- [44] A. Sobral, "Bgslibrary: An opencv c++ background subtraction library," in *IX Workshop de Viso Computacional (WVC2013)*, 2013.
- [45] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changetection.net: A new change detection benchmark dataset," in *Proc. IEEE CVPRW*, 2012, pp. 1–8.
- [46] ChangDetection.NET(CDNET), "https://www.changetection.net/."
- [47] M. J. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *CVIU*, vol. 63, no. 1, pp. 75–104, 1996.
- [48] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proc. IEEE CVPR*, 2010, pp. 2432–2439.
- [49] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE TPAMI*, vol. 33, no. 3, pp. 500–513, 2011.



Xinchen Ye received the B.S. degree from the Tianjin University, Tianjin, China, in 2006. He is currently pursuing the Ph.D. Degree at the School of Electronic Information Engineering, Tianjin University. His research interests include depth recovery and 3D imaging.



Jingyu Yang (M'10) received the B.E. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2003, and Ph.D. degree (honor) from Tsinghua University, Beijing, China, in 2009.

Since 2009, he has been with the faculty of Tianjin University, China, and is currently an Associate Professor at the School of Electronic Information Engineering. He visited Microsoft Research Asia (MSRA) from Feb.-Aug. 2011 within the MSRA's young scholar supporting program. He visited Signal

Processing Laboratory at EPFL in Lausanne, Switzerland, from July-Oct. 2012 and from Oct. 2014-Sep. 2015. His research interests mainly include image/video processing, 3D imaging, and computer vision. He was selected into the program for New Century Excellent Talents in University (NCET) from the Ministry of Education of China in 2011, and selected into the Elite Peiyang Scholar Program and Reserved Peiyang Scholar Program of Tianjin University, in 2012 and 2014, respectively.

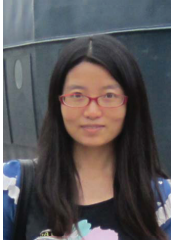


Yao Wang (M'90-SM'98-F'04) received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1983 and 1985, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara, in 1990.

She has been with the Faculty of Electrical and Computer Engineering, Polytechnic School of Engineering, New York University, Brooklyn, NY, USA, since 1990. She is the leading author of a textbook titled *Video Processing and Communications* (Prentice

Hall, 2001). Her current research interests include video coding and networked video applications, medical imaging, and pattern recognition.

Dr. Wang has served as an Associate Editor for IEEE TRANSACTIONS ON MULTIMEDIA and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. She received New York City Mayor's Award for Excellence in Science and Technology in the Young Investigator Category in year 2000. She is a co-winner of the IEEE Communications Society Leonard G. Abraham Prize Paper Award in the Field of Communications Systems in 2004. She received the Oversea Outstanding Young Investigator Award from the National Natural Science Foundation of China (NSFC) in 2005 and Yangtze River Scholar Award by the Ministry of Education of China in 2008.



Xin Sun received the B.Eng. degree from the Tianjin University, Tianjin, China, in 2012. She is currently pursuing the Master degree in Tianjin University, Tianjin, China. Her research interests include 3D imaging and computer vision.



Kun Li (M'12) received the B.E. degree in communication engineering from Beijing University of Posts and Telecommunications, Beijing, China in 2006, and Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China in 2011.

Since 2011, she has been with the faculty of Tianjin University, China, and is currently an assistant professor in the School of Computer Science and Technology of Tianjin University. Her research interests include image/video processing, image-based modeling, dynamic scene 3D reconstruction, and multi-camera imaging. She was selected into the Elite Scholar Program of Tianjin University in 2012.



Chunping Hou received the M.Eng. and Ph.D. degrees, both in electronic engineering, from Tianjin University, Tianjin, China, in 1986 and 1998, respectively.

She was a Post-Doctoral Researcher with the Beijing University of Posts and Telecommunications, Beijing, China, from 1999 to 2001. Since 1986, she has been with the faculty of the School of Electronic Information Engineering, Tianjin University, where she is currently a Full Professor and the Director of the Broadband Wireless Communications and 3-D

Imaging Institute. Her current research interests include wireless communication, 3-D image processing, and the design and applications of communication systems.