Signal Processing 168 (2020) 107362

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Spatiotemporally scalable matrix recovery for background modeling and moving object detection

Jingyu Yang[a], Wen Shi[a], Huanjing Yue[a], Kun Li[b,*], Jian Ma[b], Chunping Hou[a]

[a] School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, PR China
[b] College of Intelligence and Computing, Tianjin University, Tianjin 300072, China

ARTICLE INFO

Article history:
Received 27 February 2019
Revised 16 October 2019
Accepted 26 October 2019
Available online 28 October 2019

Keywords:
Background modeling
Motion detection
Optical flow
Matrix restoration
Video surveillance

ABSTRACT

Separating moving objects and backgrounds from a video is an important yet challenging task for video analysis due to complex moving behaviors, camera jitters/movements, and huge data amount in real-world applications. To deal with these issues, this paper proposes a unified framework called spatiotemporally scalable matrix recovery (SSMR), which has a moderate computational and space complexity scalable to temporal and spatial resolution of videos. In the proposed model, the inherent batch-mode nuclear norm for low-rank approximation is replaced with an explicitly low-rank matrix factorization in order to achieve online implementation. Motion information extracted by an optical flow method is incorporated into the data term to facilitate the separation of moving objects from the background. Affine transformation is embedded into the model and simultaneously optimized with other variables to handle camera motions. In addition, we proposed a pyramidal scheme to achieve spatial scalability for high definition videos. Experimental results demonstrate that our method outperforms many other state-of-the-art methods and can handle videos of various complex scenarios.

1. Introduction

Extracting interesting information from videos is an important yet challenging task, and has received tremendous attention [1–4]. Background modeling is one of such topics, and is useful across many applications including motion detection [5,6], object tracking [7], behavior recognition [8] and video coding [9]. Foreground-background separation is to recover background components and detect moving foreground objects from original video clips [10], and there are several challenges, such as illumination changes, obstructions, camouflage and shadows cast by the foreground objects [11] in this field. Moreover, camera moving in many scenarios leads to motion blurring and dynamic background, which is even difficult to resolve. Many research works have tackled various challenges of background/foreground separation from different perspectives.

Early research works classified pixels into either background components or foreground components by estimating the distribution of pixel intensities locally under various statistical models such as Gaussian distribution [12], the Gaussian mixture model (GMM) and its variants [13,14], the Student's-t mixture model (SMM) [15]. Being able to exploit global structure of video clips, robust principal component analysis (RPCA) [2] have been intensively investigated in background and foreground separation with various techniques such online implementation [16–18], transformation compensation [6], motion assistance [19], and spatiotemporal regularization [20,21]. Although these RPCA-based methods achieved significant progress, they still suffer from problems, such as presence of noise, lack of spatial and temporal constraints, and global motion in the background [22]. While it is difficult to solve all the challenging problems in a single algorithm, there are three key aspects for practical applications: 1) the capability in handling moving objects with complex behaviors such as slow moving and camouflages; 2) the ability to handle moving backgrounds due to camera jitter or motion, and 3) the scalability to high definition videos with affordable computational and memory resources.

To resolve the challenges mentioned above, this paper proposes a unified framework, called *spatiotemporally scalable matrix recovery* (SSMR), for background modeling and moving object detection. To prevent the complex-moving objects leaking into background components, regions of moving objects are identified by motion information via optical flow, and are then used to rectify the data consistency. The low-rank regularization of background components is relaxed to matrix factorization with explicit rank constraint, from which an online implementation is derived to

* Corresponding author.
E-mail addresses: yjy@tju.edu.cn (J. Yang), wenshi@tju.edu.cn (W. Shi), dayueer@tju.edu.cn (H. Yue), lik@tju.edu.cn (K. Li), majian@tju.edu.cn (J. Ma), hcp@tju.edu.cn (C. Hou).

https://doi.org/10.1016/j.sigpro.2019.107362
0165-1684/

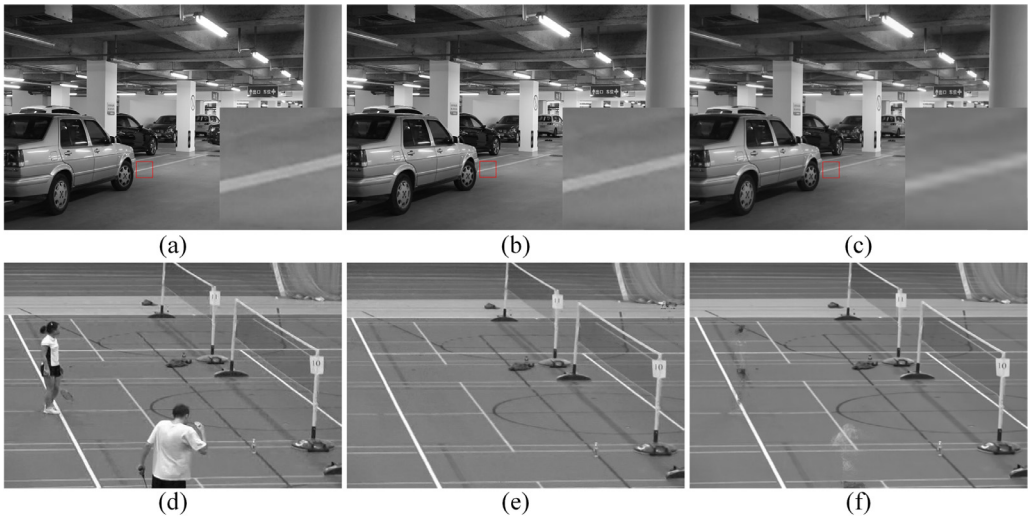

Fig. 1. The example of our proposed algorithm. The top raw is the high-resolution reconstruction result. Specifically, (a), (b), (c) are the original image, recovered background image and enlarged image by bicubic interpolation, respectively. The bottom tow is the results of dynamic videos. (d) is the input frame. (e) and (f) are results of our algorithm and the state-of-the-art method, OMoGMF [23].

avoid the computation- and memory-demanding singular value decomposition (SVD) in the batch-mode implementation. The background, foreground, and error components are updated alternatingly under the framework of the augmented Lagrangian method. To achieve spatial scalability for high definition videos, the SSMR model is applied to the lowest-frequency subband of the multiscale pyramid representation, and the recovered low-resolution background components are synthesized into high resolution version through pyramid reconstruction. To compensate background movement due to camera jitter or motion, the SSMR model is enhanced with an affine transformation, whose parameters are simultaneously estimated online along with the background, foreground, and error components such that the low-rank assumption is still valid. Experimental results on various video clips with both static and dynamic backgrounds show that the proposed method achieves the best performance among state-of-the-art online methods for both background recovery and moving object detection, and outperforms the competing batch-mode methods for most cases (Fig. 1).

This paper is an extended version of our previous work [24] and the additional contributions are summarized as below. First, we incorporate affine transformation into the model to handle videos with moving backgrounds, and derive an efficient numerical algorithm. Second, a multi-scale pyramidal scheme is proposed to achieve spatial scalability for high-definition videos. Third, this paper also provides more detailed discussions and experimental evaluations. The rest of this paper is organized as follows. Section 2 presents a brief overview on related work. In Section 3, we propose the SSMR model and numerical algorithms. Section 4 adapts the SSMR model to handle moving backgrounds by simultaneously optimization of camera motion and background/foreground components. Experimental results and analysis are given in Section 5, followed by conclusions in Section 6.

2. Related work

The separation of background and moving objects components are very challenging due to the many possible interferences such as varying illumination, slow motion of objects, and dynamic backgrounds. The fundamental rationale is to characterize their different properties in the spatiotemporal volume. Simple statistical models, *e.g.*, temporal median filtering [25] and running Gaussian average [12], are fast to detect moving objects with simple motion pattern from a static background, but are incompetent for complex motions and backgrounds with multi-modal distributions. As the most prevalent statistical model, GMM has been applied to background modeling to exploit multi-modal distributions of background and foreground intensities [13,23,26–28]. Other flexible statistical models are also investigated in the task of background modeling, such as the adaptive GMM [29], the generalized GMM [14], the Student's-t mixture model (SMM) [15], the Dirichlet process GMM [30], Gaussian conditional random field [31], and general nonparametric kernel density estimation [32]. Despite their promising results for a broad class of videos, such statistical models are applied locally, either pixel-wise or patch-wise, and inevitably ignore the global structure the spatiotemporal video volume.

Regarding global methods in background modeling, a family of low-rank matrix recovery models have been intensively investigated. In developing the robust principal component analysis (RPCA), as known as principle component pursuit, Candes et al. [2] show that background-foreground separation from videos can be formulated as a low-rank recovery problem: the background components over time are nearly duplicated and the associated matrix is theoretically rank-one; while the foreground components usually take up only a small portion of the picture, and thus can be modeled as a noise component distributed sparsely. For efficient numerical algorithm, the rank regularization of background components and sparsity regularization of foreground components are relaxed to the convex surrogates, *i.e.*, nuclear norm and ℓ_1-norm, respectively. Following this avenue, many research works have significantly improved the performance by incorporating various ingredients according to video characteristics. In [33], a two-pass RPCA strategy is proposed to refine the accuracy of foreground detection. Zhou et al. [6] proposed to detect moving objects by detecting contiguous outliers in the low-rank representation (DECOLOR), where the support of moving objects are simultaneously optimized via a Markov random field (MRF) along with background components. Cao et al. [20] enhanced RPCA by imposing spatial and temporal continuity with TV regularization. Similarly, Xin et al. [19] further imposed a smoothness prior on the foreground components besides the sparsity prior. Yang et al. [19] proposed a motion-assisted matrix restoration (MAMR) model by incorporating motion information as a weighting matrix. Similar to [19], Sobral et al. [34] constrained the foreground component by shape and confidence maps, which are both extracted from spatial saliency maps, to reinforce the foreground detection. Javed

et al. [35,36] designed an online spatiotemporal robust principal component analysis (OS-RPCA) algorithm by incorporating spatial and temporal maps into RPCA. Taehyeon et al. [37] used TUCKER2 decomposition to enhanced background subtraction, avoiding iterative process in nuclear norm minimization. In [38], Li et al. first selected a set of discriminative frame for background modeling and then extracted background with a low-rank tensor recovery model from the selected frames. More RPCA methods can be found in comprehensive surveys [22,39].

Low-rank based methods assumes that background components of video frames are well aligned otherwise the low-rankness would be violated. Videos in practical applications such as surveillance, however, usually present dynamic backgrounds due to Pan/Tilt/Zoom(PTZ) operations or shaking of cameras. RPCA and its variants methods could tolerate slight background jittering, but would fail when it comes to dynamic backgrounds. In [40], Peng et al. demonstrated that RPCA is still quite powerful in a number of vision tasks on linearly correlated images with accurate alignment. To handle dynamic backgrounds, Zhou et al. [6] simultaneously optimize the alignment and background components under the low-rank matrix recovery framework. Other similar works can be found in [41,42].

As the low-rank approximation operates on a bunch of video frames, such a batch-mode optimization not only introduces large delays, but also requires huge amount of computational and memory resources. Despite their promising performance, the heavy batch-mode background models are difficult to apply to real-time applications. By relaxing the nuclear norm to the Frobenius norm of two factorized low-rank matrices, Feng et al. [17] derived an online RPCA algorithm that significantly reduced the required computation and memory. He et al. [43] proposed an online robust Grassmannian subspace tracking algorithm that operates highly subsampled frames. Xu et al. [44] further enhanced the online Grassmannian subspace tracking with structure sparsity to provide some tolerance to homogeneous perturbations. Wang et al. [45] proposed an online expectation-maximization (EM) algorithm to solve the probabilistic robust matrix factorization model. Rodriguez and Wohlberg [18] introduced an incremental and Rank-1 modifications for thin SVD to derive an online implementation. Under the low-rank factorization framework, Yong et al. [23] modeled moving objects by a mixture of Gaussians (MoG) distribution and embedded an affine transformation operator to adapt to camera movements. In [46], Javed et al. segmented the data matrix into superpixels through graph partitioning and then utilized online maximum norm matrix decomposition and the generalized fussed lasso constraint to estimate low rank components and sparse foregrounds. Qiu et al. [47–49] designed a recursive projected compressive sensing (ReProCS) method which can successfully separate slowly changing backgrounds and foregrounds. Based on ReProCS, Narayanamurthy [50] proposed a memory-efficient robust PCA model (MERoP) with nearly-optimal memory complexity and performances guarantees.

Previous methods usually focused on one or two particular aspects of the challenges in background/foreground separation, which might limit their applicability in more practical scenario, *e.g.*, surveillance with camera panning high-definition inputs. To this end, we proposed a unified optimization framework with fourfold merits: 1) moving awareness through motion extraction, 2) affine invariance with simultaneously optimization of transformation and separation, 3) temporal scalability via online implementation, 4) spatial scalability by pyramidal implementation.

3. Spatiotemporally scalable matrix recovery

In this section, we present our SSMR method in details. Previous works [6,19] demonstrated that excluding the foreground region in the data term helps to resolve the foreground leakage problem of the plain RPCA model [2]. Therefore, we develop our model upon the batch-mode MAMR model [19], which incorporates a motion map to modulate the data consistency. We first present the temporal scalable matrix recovery model in Section 3.1, and then derive an online numerical algorithm in Section 3.2. Section 3.3 applies the model to a multiscale pyramid representation to achieve spatial scalability.

3.1. Spatiotemporally scalable matrix recovery model

Let $\mathbf{d}_k \in \mathbb{R}^{m\times 1}$ denote the vector form of the kth frame of the input video sequence consisting of m pixels. The input video sequence can be compactly represented by matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n] \in \mathbb{R}^{m\times n}$, where n is the number of frames. Denoted the background component and complementary error in $\mathbf{D}$ by $\mathbf{B}$ and $\mathbf{E}$, respectively. The RPCA model with a weighted data term is formulated as:

$$\min_{\mathbf{B},\mathbf{E}} \frac{1}{2}\|\mathbf{W}\odot(\mathbf{D}-\mathbf{B}-\mathbf{E})\|_{\mathrm{F}}^2 + \lambda_1\|\mathbf{B}\|_* + \lambda_2\|\mathbf{E}\|_1, \tag{1}$$

where $\|\cdot\|_{\mathrm{F}}$, $\|\cdot\|_*$ and $\|\cdot\|_1$ denote the Frobenius norm, nuclear norm (sum of singular values) and ℓ_1 norm of a matrix, respectively, and "$\odot$" denotes the element-wise multiplication of two matrices. λ_1 and λ_2 are the weights for the nuclear-norm term and ℓ_1-norm term. $\mathbf{W}\in[0,\,1]^{m\times n}$ is a weighting matrix to exclude occluded areas by moving objects, where the additive observation model is violated. The weighting matrix is constructed from motion information estimated by optical flow [51]. Denote by $\left(\mathbf{o}_k^{\mathrm{x}}, \mathbf{o}_k^{\mathrm{y}}\right)$ the horizontal component and vertical component of the optical flow, respectively, estimated between the current frame $\mathbf{d}_k$ and its anchor frame. The elements of $\mathbf{W}$ are constructed as follows:

$$w_{ik} = \begin{cases} 0, & \sqrt{\left(\mathrm{o}_{\mathrm{ik}}^{\mathrm{x}}\right)^2 + \left(\mathrm{o}_{\mathrm{ik}}^{\mathrm{y}}\right)^2} \geq \mathrm{t}, \\ 1, & \text{otherwise}, \end{cases} \tag{2}$$

where t is a threshold to control the level of optical-flow intensity regarded as motion.

The batch-mode model (1) could be overwhelmed by huge data of high-definition video applications. However, the optimization of $\|\mathbf{B}\|_*$ is inherently in the batch manner, which is difficult to derive a temporally scalable solution. Inspired by Feng et al. [17], the nuclear norm $\|\mathbf{B}\|_*$ can be replaced by the following equivalent form with an explicit low-rank matrix factorization

$$\|\mathbf{B}\|_* = \inf_{\mathbf{L},\mathbf{C}} \left\{ \frac{1}{2}\|\mathbf{L}\|_{\mathrm{F}}^2 + \frac{1}{2}\|\mathbf{C}\|_{\mathrm{F}}^2 : \mathbf{B} = \mathbf{L}\mathbf{C}^\top \right\}, \tag{3}$$

where $\mathbf{L}\in\mathbb{R}^{m\times r}$ and $\mathbf{C}\in\mathbb{R}^{n\times r}$ are the two factorized matrices of $\mathbf{B}$ with the rank r smaller than both m and n. From a perspective of signal synthesis, $\mathbf{L}$ could be interpreted as the low-dimensional subspace basis where the background $\mathbf{B}$ lies while $\mathbf{C}$ is the corresponding coefficients under the basis. Then, model (1) is reformulated as follows:

$$\min_{\mathbf{L},\mathbf{C},\mathbf{E}} \frac{1}{2}\left\|\mathbf{W}\odot(\mathbf{D}-\mathbf{L}\mathbf{C}^\top-\mathbf{E})\right\|_{\mathrm{F}}^2 + \frac{\lambda_1}{2}(\|\mathbf{L}\|_{\mathrm{F}}^2 + \|\mathbf{C}\|_{\mathrm{F}}^2) + \lambda_2\|\mathbf{E}\|_1. \tag{4}$$

To derive an online algorithm, the objective function should be decoupled in terms of observed data $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n$. To this end, we reformate the objective function in Eq. (4) into the following form:

$$\begin{aligned} \mathrm{f}_n(\mathbf{L},\mathbf{C},\mathbf{E};\mathbf{D}) &\triangleq \sum_{k=1}^{n} \ell(\mathbf{c}_k,\mathbf{e}_k,\mathbf{L};\mathbf{d}_k) + \frac{\lambda_1}{2}\|\mathbf{L}\|_{\mathrm{F}}^2, \\ \ell(\mathbf{c}_k,\mathbf{e}_k,\mathbf{L};\mathbf{d}_k) &\triangleq \min_{\mathbf{c}_k,\mathbf{e}_k} \frac{1}{2}\|\mathbf{w}_k\odot(\mathbf{d}_k-\mathbf{L}\mathbf{c}_k-\mathbf{e}_k)\|_2^2 \\ &\quad + \frac{\lambda_1}{2}\|\mathbf{c}_k\|_2^2 + \lambda_2\|\mathbf{e}_k\|_1, \end{aligned} \tag{5}$$

where $\mathbf{w}_k$ denotes the weighting map for frame $\mathbf{d}_k$, $\mathbf{c}_k \in \mathbb{R}^{r\times 1}$ represents the coefficient of $\mathbf{d}_k$ under basis $\mathbf{L}$, and $\mathbf{e}_k \in \mathbb{R}^m$ is the error component in the input frame. Given the same data matrix $\mathbf{D}$, minimization of (5) is equivalent to the solution of (4). However, the formulation in (5) allows frame-wise alternating minimization of $(\mathbf{c}_k, \mathbf{e}_k, \mathbf{L})$ as a stochastic optimization [17], which is detailed as follows.

3.2. Numerical algorithm

With the incoming frame $\mathbf{d}_k$ and the associated binary weights, we are to solve the coefficient $\mathbf{c}_k$, sparse error $\mathbf{e}_k$ of $\mathbf{d}_k$ and update the background basis $\mathbf{L}_k$ given their counterparts associated with the last frame $\mathbf{d}_{k-1}$. Noting that simultaneous optimization of these variables are difficult and impractical, we resort to the alternate direction method (ADM). Let l denote the iteration index of the alternating optimization. The coefficient $\mathbf{c}_k^{l+1}$ and sparse error $\mathbf{e}_k^{l+1}$ are updated in turn by minimizing the following sub-problems.

$$\begin{cases} \min_{\mathbf{c}} \frac{1}{2}\left\|\mathbf{w}_k \odot (\mathbf{d}_k - \mathbf{L}_{k-1}\mathbf{c} - \mathbf{e}_k^l)\right\|_2^2 + \frac{\lambda_1}{2}\|\mathbf{c}\|_2^2 \\ \min_{\mathbf{e}} \frac{1}{2}\left\|\mathbf{w}_k \odot (\mathbf{d}_k - \mathbf{L}_{k-1}\mathbf{c}_k^{l+1} - \mathbf{e})\right\|_2^2 + \lambda_2\|\mathbf{e}\|_1 \end{cases} \tag{6}$$

For the initialization of alternating optimization, we have $\mathbf{c}_k^0 = \mathbf{c}_{k-1}$ and $\mathbf{e}_k^0 = \mathbf{e}_{k-1}$. $\mathbf{c}_k^{l+1}$-subproblem is solved by least squares:

$$\mathbf{c}_k^{l+1} = (\mathbf{L}_{k-1}^\top \mathbf{S}_1 \mathbf{L}_{k-1} + \lambda_1 \mathbf{I}_r)^{-1} \mathbf{L}_{k-1}^\top \mathbf{S}_1 (\mathbf{d}_k - \mathbf{e}_k^l), \tag{7}$$

where $\mathbf{S}_1 \triangleq \mathrm{diag}(\mathbf{w}_k \odot \mathbf{w}_k)$ and $\mathrm{diag}(\mathbf{x})$ is an operator to consgtruct a diagonal matrix with $\mathbf{x}$ as its diagonal elements. $\mathbf{I}_r$ is a $r \times r$ identity matrix. For compact notation, let $\mathbf{b}_k \triangleq \mathbf{d}_k - \mathbf{L}_{k-1}\mathbf{c}_k$. Then, $\mathbf{e}_k^{l+1}$-subproblem has the following closed-form shrinkage solution:

$$\mathbf{e}_k^{l+1} = \mathrm{shrink}(\mathbf{d}_k - \mathbf{L}_{k-1}\mathbf{c}_k^{l+1}, \lambda_2 ./ \mathbf{w}_k) \tag{8}$$

where "./" denotes element-wise division, and $\mathrm{S}(\mathbf{x}, \boldsymbol{\tau})$ is element-wise shrinkage function defined as

$$\mathrm{S}(x_i, \tau_i) \triangleq \begin{cases} 0, & |x_i| < \delta_i, \\ \mathrm{sign}(x_i)(|x_i| - \tau_i), & \text{otherwise.} \end{cases} \tag{9}$$

When the alternating optimization is convergent at iteration l^*, we have $\mathbf{c}_k = \mathbf{c}_k^{l^*}$ and $\mathbf{e}_k = \mathbf{e}_k^{l^*}$.

Model (5) shows that the basis $\mathbf{L}$ is updated by minimizing the cumulative loss with the previous data $\{\mathbf{d}_k, \mathbf{c}_k, \mathbf{e}_k\}$. For concise notation, define the observation matrix, weighting matrix, coefficient matrix, and error matrix as

$$\begin{aligned} \mathbf{D}_k &\triangleq [\mathbf{d}_1, \ldots, \mathbf{d}_k], \\ \mathbf{W}_k &\triangleq [\mathbf{w}_1, \ldots, \mathbf{w}_k], \\ \mathbf{C}_k^\top &\triangleq [\mathbf{c}_1, \ldots, \mathbf{c}_k], \\ \mathbf{E}_k &\triangleq [\mathbf{e}_1, \ldots, \mathbf{e}_k]. \end{aligned} \tag{10}$$

The sub-problem to update $\mathbf{L}$ is formulated as following minimization problem.

$$\min_{\mathbf{L}} \frac{1}{2}\left\|\mathbf{W}_k \odot (\mathbf{D}_k - \mathbf{L}\mathbf{C}_k^\top - \mathbf{E}_k)\right\|_{\mathrm{F}}^2 + \frac{\lambda_1}{2}\|\mathbf{L}\|_{\mathrm{F}}^2. \tag{11}$$

Define the previously recovered background frames as $\mathbf{R}_k \triangleq \mathbf{D}_k - \mathbf{E}_k$, and introduce a new matrix $\mathbf{Y}_k \triangleq \mathbf{R}_k - \mathbf{L}\mathbf{C}_k^\top$. Minimization (11) can be transformed into the following form:

$$\min_{\mathbf{Y},\mathbf{L}} \|\mathbf{W}_k \odot \mathbf{Y}\|_{\mathrm{F}}^2 + \lambda_1\|\mathbf{L}\|_{\mathrm{F}}^2 + \lambda_3\left\|\mathbf{Y} - \mathbf{R}_k + \mathbf{L}\mathbf{C}_k^\top\right\|_{\mathrm{F}}^2. \tag{12}$$

where λ_3 is a penalty weight. We update $\mathbf{Y}_k$ and $\mathbf{L}_k$ alternatively. Let l denote the iteration index of the alternating optimization, and initialize alternating procedure as $\mathbf{Y}_k^0 = \mathbf{Y}_{k-1}$, $\mathbf{L}_k^0 = \mathbf{L}_{k-1}$. $\mathbf{Y}_k^{l+1}$ is solved by least squares:

$$\mathbf{Y}_k^{l+1} = 2\lambda_3(\mathbf{R}_k - \mathbf{L}_k^l \mathbf{C}_k^\top)./(2\lambda_3 + \mathbf{W}_k \odot \mathbf{W}_k). \tag{13}$$

Similarly, the background basis has the following least-squares closed-form solution:

$$\mathbf{L}_k^{l+1} = \lambda_3(\mathbf{R}_k - \mathbf{Y}_k^{l+1})\mathbf{C}_k(\lambda_3\mathbf{C}_k^\top\mathbf{C}_k + \lambda_1\mathbf{I})^{-1}. \tag{14}$$

Let $\mathbf{P}_k \triangleq \mathbf{R}_k\mathbf{C}_k = \sum_{t=1}^k \mathbf{r}_t\mathbf{c}_t^\top \in \mathbb{R}^{m\times r}$, $\mathbf{X}_k \triangleq \mathbf{Y}_k\mathbf{C}_k = \sum_{t=1}^k \mathbf{y}_t\mathbf{c}_t^\top \in \mathbb{R}^{m\times r}$ and $\mathbf{Z}_k \triangleq \mathbf{C}_k^\top\mathbf{C}_k = \sum_{t=1}^k \mathbf{c}_t\mathbf{c}_t^\top \in \mathbb{R}^{r\times r}$. The update of $\mathbf{L}_k^{l+1}$ is further simplified as:

$$\mathbf{L}_k^{l+1} = \lambda_3(\mathbf{P}_k - \mathbf{X}_k^{l+1})(\lambda_3\mathbf{Z}_k + \lambda_1\mathbf{I})^{-1}, \tag{15}$$

where matrices $\mathbf{P}_k$, $\mathbf{X}_k$, and $\mathbf{Z}_k$ are updated incrementally:

$$\begin{aligned} \mathbf{P}_k &= \mathbf{P}_{k-1} + \mathbf{r}_k\mathbf{c}_k^\top, \\ \mathbf{X}_k^{l+1} &= \mathbf{X}_{k-1} + \mathbf{y}_k^{l+1}\mathbf{c}_k^\top, \\ \mathbf{Z}_k &= \mathbf{Z}_{k-1} + \mathbf{c}_k\mathbf{c}_k^\top. \end{aligned} \tag{16}$$

In this way, the numerical algorithm for model (3) is temporally scalable as the required computation and memory are not increasing with the number of incoming frames. The numerical algorithm is summarized into Algorithm 1.

Algorithm 1 SSMR algorithm.

Input: observed data $[\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K] \in \mathbb{R}^{m\times K}$, current basis $\mathbf{L}_0 \in \mathbb{R}^{m\times r}$, parameters $\lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}$, number of frames K
Initialize: motion map $\mathbf{w}_1 = \mathbf{1} \in \mathbb{R}^m$, coefficient $\mathbf{c}_1 = \mathbf{0} \in \mathbb{R}^r$, sparse error $\mathbf{e}_1 = \mathbf{0} \in \mathbb{R}^m$
for k = 1 to K **do**
  Construct motion map $\mathbf{w}_k$ by Eq. (2):
  //Line 6–9 solve problem $\mathbf{c}_k$ and $\mathbf{e}_k$
  while not converged **do**
    Update the coefficient $\mathbf{c}_k$ by Eq. (7):
    Update the sparse error $\mathbf{e}_k$ by Eq. (9):
  end while
  //Line 11–14 solve variable $\mathbf{Y}_k$ and problem $\mathbf{L}_k$
  while not converged **do**
    Update $\mathbf{Y}_k$ by Eq. (13):
    Update the basis $\mathbf{L}_k$ by Eq. (15):
  end while
end for
Output: Low-rank data matrix $\mathbf{B}_K = \mathbf{L}_K\mathbf{C}_K^\top$, Sparse error matrix $\mathbf{E}_K = [\mathbf{e}_1, \ldots, \mathbf{e}_K]$

3.3. Pyramidal scheme for spatial scalability

We note that it is difficult to directly process high-definition videos even with the online Algorithm 1. The most straightforward way is to down sample the input video to reduce the computation and memory requirement [23]. However, directly down-sampling tends to lose high resolution information. As shown in Fig. 2, we propose to apply our model on the pyramid to separate the foreground components at low-resolution scale and recover the background components at high-resolution through pyramid reconstruction. We choose the framing pyramid [52] for its promising capability in noise resilience.

Let J be the number of scales. The video frame $\mathbf{d}_k$ is decomposed into a J-scale pyramid, generating a low-resolution approximation denoted by $\mathbf{d}_k^0$, and J high-frequency subbands denoted by $[\mathbf{h}_k^1, \mathbf{h}_k^2, \ldots, \mathbf{h}_k^J]$ from coarse to fine. Due to successive down-sampling, $\mathbf{d}_k^0$ has only $1/4^J$ the number of pixels as $\mathbf{d}_k$. To save computation and memory resource, we perform our SSMR algorithm on $\mathbf{d}_k^0$ to obtain a coarse version background $\mathbf{b}_k^0$ and the corresponding foreground mask $\mathbf{m}_k^0$ via background subtract. A high-resolution background is recovered by adding back details

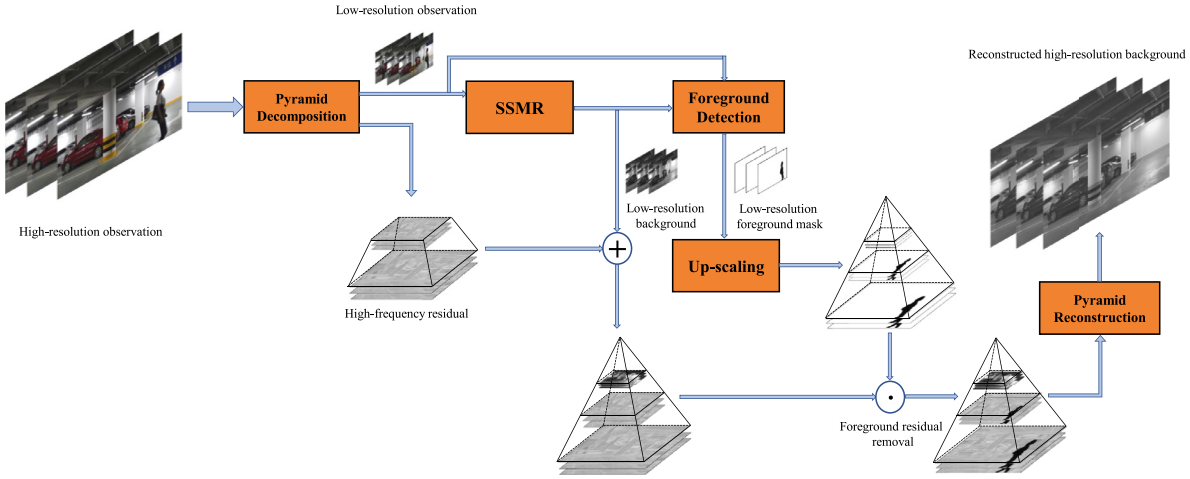

Fig. 2. The work flow of the multi-resolution representation.

from high-frequency subbands $[\mathbf{h}_k^1, \mathbf{h}_k^2, \ldots, \mathbf{h}_k^J]$ through inverting the pyramid. However, high-frequency subbands containing foreground information would bring in foreground residues. To avoid these interference, we enlarge the foreground mask $\mathbf{m}_k^0$ by bicubic interpolation into a mask pyramid $[\mathbf{m}_k^1, \mathbf{m}_k^2, \ldots, \mathbf{m}_k^J]$ associated with pyramid $[\mathbf{b}_k^0, \mathbf{h}_k^1, \mathbf{h}_k^2, \ldots, \mathbf{h}_k^J]$. Then, foreground information in subband $\mathbf{h}_k^j$ is removed using the mask: $\tilde{\mathbf{h}}_k^j = \mathbf{h}_k^j \odot \mathbf{m}_k^j$. Finally, the high resolution background $\mathbf{b}_k^h$ is reconstructed from pyramid $[\mathbf{b}_k^0, \tilde{\mathbf{h}}_k^1, \tilde{\mathbf{h}}_k^2, \ldots, \tilde{\mathbf{h}}_k^J]$.

4. Robust SSMR against moving backgrounds

In the proposed SSMR model (1), the low-rank prior on background components suggest that backgrounds in video clips are static at least approximately. However, cameras could have jitters and motions. For example, outdoor surveillance cameras could be shaken in bad weather such as fierce wind or huge waves, resulting in videos with dynamic background; videos shot by a handheld camera also have varying global motions. Such videos with camera jitter and motion violate the low-rank assumption for background modeling, leading to severe blurring artifacts in recovered backgrounds. To achieve robustness to moving backgrounds, we extend our SSMR model by embedding a transformation operator, which is simultaneously optimized with other variables.

4.1. SSMR with online alignment

To align the video frames, a reference frame (usually the first frame) is selected as the benchmark view and the subsequent frames are warped to the viewpoint of the reference frame, so that the low-rank assumption is still hold. Moving backgrounds are mainly due to camera motions such as jitter, rotation, translation, or/and panning, which could be described by affine transformations. Denote by $\tau_k \in \mathbb{R}^p$ the affine transform for frame $\mathbf{d}_k$, and by $\mathbf{d}_k \circ \tau_k$ the aligned version of $\mathbf{d}_k$ to the reference frame, where $\circ$ is the transform operator. Then, $\mathbf{D} \circ \boldsymbol{\tau} \triangleq [\mathbf{d}_1 \circ \tau_1, \mathbf{d}_2 \circ \tau_2, \ldots, \mathbf{d}_n \circ \tau_n]$ is the observed matrix after the transformation. The most straightforward way is to first estimate the transformation and then perform matrix recovery on the aligned video. Instead, we simultaneously optimize the transformation and the background, yielding the following model.

$$\min_{\mathbf{L},\mathbf{C},\mathbf{E},\boldsymbol{\tau}} \frac{1}{2}\left\|\mathbf{W} \odot (\mathbf{D} \circ \boldsymbol{\tau} - \mathbf{L}\mathbf{C}^\top - \mathbf{E})\right\|_{\mathrm{F}}^2 + \frac{\lambda_1}{2}(\|\mathbf{L}\|_{\mathrm{F}}^2 + \|\mathbf{C}\|_{\mathrm{F}}^2) + \lambda_2\|\mathbf{E}\|_1 + \lambda_4\|\mathbf{W} \odot (\mathbf{D} \circ \boldsymbol{\tau} - \mathbf{F})\|_1, \tag{17}$$

where $\mathbf{F} \triangleq [\mathbf{f}, \mathbf{f}, \ldots, \mathbf{f}]$ denotes the reference matrix constructed by repeating the reference frame $\mathbf{f}$ (vector form) as columns, and the last term of the objective function regularizes the alignment error. λ_4 signifies the regularization coefficient.

Similar to Model (4), the above model is decoupled in terms of $\{\mathbf{d}_k, \mathbf{c}_k, \mathbf{e}_k, \tau_k\}$. Given the basis from last update $\mathbf{L}_{k-1}$, and the incoming frame $\mathbf{d}_k$ and its associated weighting map $\mathbf{w}_k$, we have the following subproblems to solve under the alternating optimization framework.

$$\begin{cases} \min_\tau \ \phi(\tau, \mathbf{c}_k^l, \mathbf{e}_k^l) + \lambda_4\|\mathbf{w}_k \odot (\mathbf{d}_k \circ \tau - \mathbf{f})\|_1, \\ \min_{\mathbf{c}} \ \phi(\tau_k^{l+1}, \mathbf{c}, \mathbf{e}_k^l) + \frac{\lambda_1}{2}\|\mathbf{c}\|_2^2, \\ \min_{\mathbf{e}} \ \phi(\tau_k^{l+1}, \mathbf{c}_k^{l+1}, \mathbf{e}) + \lambda_2\|\mathbf{e}\|_1, \end{cases} \tag{18}$$

where, for concise presentation, we define the data term as

$$\phi(\tau, \mathbf{c}, \mathbf{e}) \triangleq \frac{1}{2}\|\mathbf{w}_k \odot (\mathbf{d}_k \circ \tau - \mathbf{L}_{k-1}\mathbf{c} - \mathbf{e})\|_2^2. \tag{19}$$

As $\mathbf{d}_k \circ \tau$ is nonlinear with respect to the geometric transformation, the τ-subproblem does not have a closed form solution. So we use an incremental refinement strategy which approximates $\mathbf{d}_k \circ \tau$ by a linear expansion $\mathbf{d}_k \circ \tau \approx \mathbf{d}_k \circ \tau_k^l + \mathbf{J}\triangle\tau$, where $\mathbf{J} \in \mathbb{R}^{m \times p}$ is the Jacobian matrix of $\mathbf{d}_k \circ \tau_k^l$ with respect to τ_k^l. Let $\mathbf{x}$ denotes the coordinate set of $\mathbf{d}_k \circ \tau_k^l$ after transformation, and $\mathbf{J}$ has the following formulation:

$$\mathbf{J} = \frac{\partial(\mathbf{d_k} \circ \zeta)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \zeta}\bigg|_{\zeta = \tau_{\mathbf{k}}^{\mathbf{l}}} \tag{20}$$

where $\frac{\partial(\mathbf{d}_k \circ \zeta)}{\partial \mathbf{x}}$ and $\frac{\partial \mathbf{x}}{\partial \zeta}$ denote the gradient field of $\mathbf{d}_k \circ \tau_k^l$ and the Jacobian matrix of the coordinate set $\mathbf{x}$ with respect to τ_k^l, respectively. Specifically, assume the pixel q and its coordinate (i, j) in $\mathbf{d}_k$, so the qth row of Jacobian matrix $\mathbf{J}$ can be formulate as [53]:

$$\mathbf{J}_q = \begin{bmatrix} \nabla_x \mathbf{d}_k \circ \tau_k^l & \nabla_y \mathbf{d}_k \circ \tau_k^l \end{bmatrix} \begin{bmatrix} 1 & 0 & i & j & 0 & 0 \\ 0 & 1 & 0 & 0 & i & j \end{bmatrix} \tag{21}$$

where $\nabla_x \mathbf{d}_k \circ \tau_k^l$ and $\nabla_y \mathbf{d}_k \circ \tau_k^l$ denote the horizontal and the vertical gradient of pixel q, respectively.

Instead of solving τ directly, we solve $\triangle\tau$ by approximating $\phi(\tau, \mathbf{c}, \mathbf{e})$ with the linear expansion:

$$\phi(\triangle\tau, \mathbf{c}_k^l, \mathbf{e}_k^l) \approx \frac{1}{2}\|\mathbf{w}_k \odot (\mathbf{J}\triangle\tau - \mathbf{b}_1)\|_2^2, \quad \mathbf{b}_1 \triangleq \mathbf{L}_{k-1}\mathbf{c}_k^l + \mathbf{e}_k^l - \mathbf{d}_k \circ \tau_k^l. \tag{22}$$

Moreover, the ℓ_1-norm term is solved by the weighted least squares in alternating iterations. Then, the $\triangle\tau$-subproblem is

transformed into the following form.

$$\min_{\triangle\tau} \phi(\triangle\tau, \mathbf{c}_k^l, \mathbf{e}_k^l) + \lambda_4 \|\mathbf{v} \odot \mathbf{w}_k \odot (\mathbf{J}\triangle\tau - \mathbf{b}_2)\|_2^2,$$
$$\mathbf{b}_2 \triangleq \mathbf{f} - \mathbf{d}_k \circ \tau_k^l. \tag{23}$$

The weighting vector $\mathbf{v}$ is constructed as $v(i) = 1/\sqrt{|a(i)| + \epsilon}$, where $\mathbf{a} \triangleq \mathbf{w}_k \odot (\mathbf{J}\triangle\tau - \mathbf{b}_2)$ and ϵ is a small constant to avoid division by zeros.

Then, $\triangle\tau_k^{l+1}$-subproblem has the following closed-form by least squares.

$$\triangle\tau_k^{l+1} = \left(\mathbf{J}^\top(\mathbf{S}_1 + 2\lambda_4\mathbf{S}_2)\mathbf{J}\right)^{-1}\mathbf{J}^\top\left(\mathbf{S}_1\mathbf{b}_1 + 2\lambda_4\mathbf{S}_2\mathbf{b}_2\right),$$
$$\mathbf{S}_1 \triangleq \mathrm{diag}(\mathbf{w}_k \odot \mathbf{w}_k),$$
$$\mathbf{S}_2 \triangleq \mathrm{diag}(\mathbf{v} \odot \mathbf{w}_k \odot \mathbf{v} \odot \mathbf{w}_k), \tag{24}$$

where $\mathrm{diag}(\mathbf{x})$ returns the diagonal matrix with $\mathbf{x}$ as its diagonal elements. Transformation operator τ_k is updated by $\tau_k^{l+1} \leftarrow \tau_k^l + \triangle\tau_k^{l+1}$.

Similarly, $\mathbf{c}_k^{l+1}$-subproblem also has a least squares solution:

$$\mathbf{c}_k^{l+1} = \left(\mathbf{L}_{k-1}^\top\mathbf{S}_2\mathbf{L}_{k-1} + \lambda_1\mathbf{I}\right)^{-1}\mathbf{L}_{k-1}^\top\mathbf{S}_2(\mathbf{d}_k \circ \tau_k^{l+1} - \mathbf{e}_k^l) \tag{25}$$

Given $\triangle\tau_k^{l+1}$ and $\mathbf{c}_k^{l+1}$, $\mathbf{e}_k^{l+1}$-subproblem is a standard ℓ_1-norm minimization with the following shrinkage solution.

$$\mathbf{e}_k^{l+1} = \mathrm{shrink}\left(\mathbf{d}_k \circ \tau_k^{l+1} - \mathbf{L}_{k-1}\mathbf{c}_k^{l+1}, \lambda_2./\mathbf{w}_k\right). \tag{26}$$

The alternating minimization procedure is repeated until convergence. With $\{\mathbf{C}_k, \mathbf{E}_k, \boldsymbol{\tau}_k\}$, the background basis $\mathbf{L}$ is updated as in Algorithm 1 (Formulas (13)∼(15)) except that observed frames $\{\mathbf{d}_k\}$ are now replaced with their aligned version $\{\mathbf{d}_k \circ \tau_k\}$. The overall algorithm to solve our robust SSMR model with simultaneous alignment is summarized in Algorithm 2.

Algorithm 2 RSSMR algorithm.

Input: observed data $[\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K] \in \mathbb{R}^{m\times K}$, current basis $\mathbf{L}_0 \in \mathbb{R}^{m\times r}$, parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in \mathbb{R}$, number of frames K
Initialize: motion map $\mathbf{w}_1 = \mathbf{1} \in \mathbb{R}^m$,
coefficient $\mathbf{c}_1 = \mathbf{0} \in \mathbb{R}^r$,
sparse error $\mathbf{e}_1 = \mathbf{0} \in \mathbb{R}^m$,
transform parameter $\tau_1 = \mathbf{0} \in \mathbb{R}^p$
for k = 1 to K **do**
  Construct motion map $\mathbf{w}_k$ by Eq. (2)
  //Line 6–13 solve problem $\triangle\tau$, $\mathbf{c}_k$ and $\mathbf{e}_k$
  while not converged **do**
    Estimate the Jacobian matrix: $\mathbf{J} = \frac{\partial \mathbf{d}_k}{\partial \tau}|_{\tau=\tau_k^l}$
    Update $\triangle\tau$ by Eq. (24)
    Update $\mathbf{c}_k$ by Eq. (25)
    Update the sparse error $\mathbf{e}_k$ by Eq. (26)
    $\tau_k^{l+1} = \tau_k^l + \triangle\tau_k^{l+1}$
    Transform $\mathbf{d}_k$ corresponding to τ_k^{l+1}
  end while
  //Line 15–18 solve variable $\mathbf{Y}_k$ and problem $\mathbf{L}_k$
  while not converged **do**
    Update $\mathbf{Y}_k$ by Eq. (13):
    Update the basis $\mathbf{L}_k$ by Eq. (15):
  end while
end for
Output: Low-rank data matrix $\mathbf{B}_K = \mathbf{L}_K\mathbf{C}_K^\top$,
Sparse error matrix $\mathbf{E}_K = [\mathbf{e}_1, \ldots, \mathbf{e}_K]$

4.2. Implementation issues

We note that strong camera motion could cause global blurring for the captured frames. Since aligning blurred frames is not as accurate as clear ones, blurring artifacts would leak into the recovered background components. One can remove blurring artifacts using advanced deblurring techniques before geometrical alignment. However, blind motion deblurring itself is still a challenging problem. Instead, we propose to detect blurring frames and exclude them from the subsequent processing. The exclusion of blurred frames does not affect the overall task as wide camera motion only occur short periods in most applications. As shown in Fig. 3, sharp images have strong gradient fields while their blurred version have weaker ones. Therefore, we detect blurred images using the total variation on their gradient fields. Let $\nabla_x\mathbf{d}_k$ and $\nabla_y\mathbf{d}_k$ be the horizontal and vertical gradient component the $\mathbf{k}$th frame. The total variation for frame $\mathbf{d}_k$ is defined as:

$$TV_k = \sum_{i=1}^{m}\sqrt{\nabla_x^2\mathbf{d}(i) + \nabla_y^2\mathbf{d}(i)}. \tag{27}$$

If the total variation TV_k is smaller than a threshold T, frame $\mathbf{d}_k$ is considered as a blurred frame and excluded from the background modelling.

5. Experimental results

In this section, we evaluate the proposed method on static, dynamic and high-definition videos, respectively. We first introduce competing methods, test video sequences, and performance metrics in Section 5.1. The setting of experiments is presented in Section 5.2. Our algorithm is compared with several state-of-the-art methods on both background recovery and foreground detection on videos with static backgrounds in Section 5.3 and those with dynamic backgrounds in Section 5.4. The comparison results of high-definition videos are presented in Section 5.5, followed by discussion on computational complexity in Section 5.6.

5.1. Competing methods, datasets, and performance metrics

In this paper, we compare with ten prominent methods, including four batch-mode methods, *i.e.*, DECOLOR [6], RPCA [2], TVR-PCA [20], MAMR [19], and six online-mode methods: ORPCA [17], t-GRASTA [54], OMoGMF [23], incPCP-TI [42], MEROP [50], ReProCS [49]. The source codes of all the competing methods are downloaded from their project websites.

To evaluate the effectiveness of the proposed method, we test our algorithm on the CDnet2014 dataset [55] and SBIdataset [56]. CDnet2014 dataset [55] includes 11 video categories and 53 sequences, while SBIdataset [56] contains 14 typical sequences. These video clips range over various challenges such as illumination changes, slowly moving objects, and occlusion. Due to the space limit, we depict the results of some representative sequences, including eleven challenging video clips selected from CDnet2014 [55] (*Sofa* and *WinterDriveway* with intermittent object motion, *Office, Highway* and *PETS2006* in baseline, *PeopleInShade* and *Bungalows* with shadow, *Badminton, Boulevard, Sidewalk* and *Traffic* with camera jitter), three sequences *Board, Candela* and *Hallmonitor* from SBIdataset [56] and other two classic video sequences named *Cars* and *Browser1*. In addition, for evaluation on high-definition video sequences, we shot four video clips (*Garage1, Garage3, Hall*, and *Campus*) by Canon 60D. All the competing methods are operated on the original video frames except for MEROP [50] and ReProCS [49], because the update of their parameters causes the out of memory problem in our PC. Therefore, we downsample the test frames by a factor of four, and upsample separated backgrounds and foregrounds for comparison.

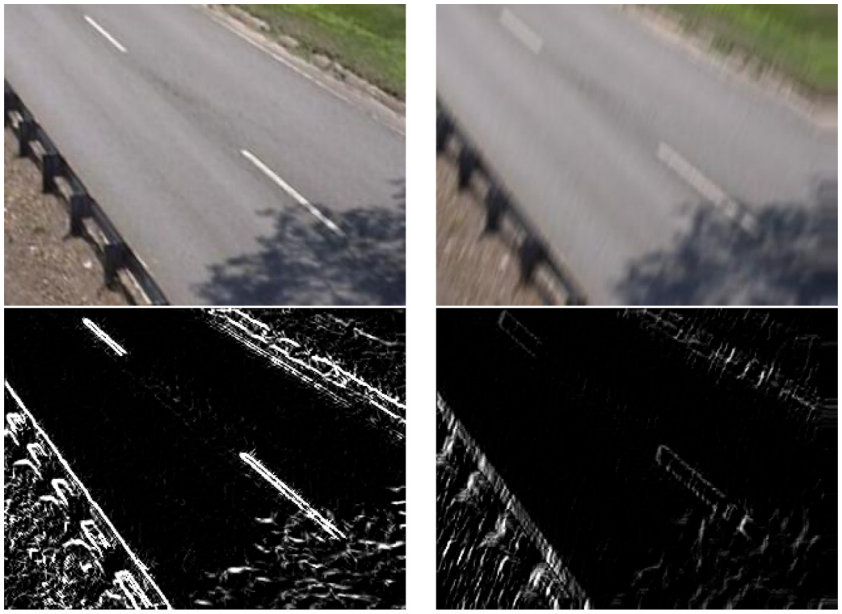

Fig. 3. The example of gradient field. (a) clean image and its gradient field, (b) blur image and its gradient field.

Two commonly-used metrics, peak-to-noise ratio (PSNR) and structural similarity index (SSIM), are used for quantitative evaluation of recovered static backgrounds. *F*-measure is employed for the evaluation of moving object detection. Let N_{cf}, N_f, and N_{gt} denote the number of pixels in the correctly-detected region, detected region, and ground-truth region of moving objects, respectively. With the definition of *precision* $P \triangleq N_{cf}/N_f$ and *recall* $R \triangleq N_{cf}/N_{gt}$. The *F*-measure metric is defined as $F_m \triangleq 2PR/(P+R)$. In fact, the *F*-measure balances the recall and precision and gives an overall quantitative evaluation. A larger F_m value suggests more accurate foreground detection.

5.2. Experiment setting

For all comparison methods, we use the default parameters. For regularization parameters of our method, we set $\lambda_1 = 0.1/\sqrt{m}$, $\lambda_2 = 1$, $\lambda_3 = 1/\sqrt{m}$, $\lambda_4 = 10$. λ_1 is inversely proportional to m, the number of pixels in a frame, as suggested in most literature on low-rank matrix recovery [16]. $\lambda_2 = 1$ is a moderate setting as a small value would result in background leakage while a too large value would lead to foreground leakage. λ_3 is the regularization parameter to ensure accurate results. λ_4 is set at a relative large value to ensure accurate alignment. The convergence tolerance is set at $10e^{-4}$. The threshold value t for motion map generation from optical flow is set at 0.5. For all online RPCA based methods, we initialize the subspace basis via randomly selecting a mini-batch of original frames. Empirically, the rank of low-dimensional subspace is set at 5, *i.e.*, the size of L is $m \times 5$.

For videos with static backgrounds, consecutive 200 frames are used for experiments. For videos with moving backgrounds, we exclude those frames with severe blurring artifacts, and 120 frames are used for evaluation.

5.3. Results on videos with static backgrounds

5.3.1. Background recovery

Tables 1 and 2 show the PSNR and SSIM results for background recovery, respectively. The best result for each test clip is in bold and the second highest value is underlined. Sequences from different datasets are separated by a dash line for convenient observation. Our SSMR method achieves almost the same average performance as the best batch-mode method MAMR [19]: only 0.55 dB PSNR loss and 0.0008 SSIM loss, which demonstrates that the proposed model does not scarifies performance with appealing spatiotemporal scalability. Particularly, our method significantly outperforms other five online approaches. We also note that MAMR [19] has obvious higher PSNR results than our model for most sequences as the background recovery of a frame in the batch mode exploits both the backward and forward temporal correlations while our online model uses only previous frames. But the results are perceptual close as SSIM differences are within 0.001, which is also verified in the following visual comparison.

Fig. 4 presents visual results of recovered backgrounds by our method and competing ones. Background recovered by our method are more closer to the ground truth, while those recovered by others present foreground leakage or smearing artifacts (marked in the red box) for most cases. For example, in *WinterDriveWay*, backgrounds recovered by RPCA [2], TVRPCA [20], OMoGMF [23], ORPCA [17], incPCP-TI [42], MEROP [50] and ReProCS [49] suffer from smearing and ghosting artifacts because the slowly-moving car is mistreated as part of the background component. Similarly phenomena could be also observed in the results of *Browser, Hallmonitor, Office, Candela* and *Sofa*, where there are objects are stationary for a while besides moving. Moreover, moving objects of large areas violate the sparse-foreground assumption in RPCA-type models, and thus results in severe foreground leakage. For example, the background of *Bungalows* recovered by OMoGMF [23], RPCA [2] and TVRPCA [20] contain considerable amount of noise in the region of car. The man in *Board* hanging out in front of the white board severely leaks into the results recovered by most methods except for MAMR [19] and our proposed SSMR model.

Both the quantitative and visual results demonstrate that the proposed SSMR does not only provide spatiotemporal scalability but also ensures high performance background performance, particularly in handling challenging cases such as slowly-moving objects, occlusion and illumination.

Table 1
PSNR comparison of background recovery with different methods for videos with static background.

	Batch-mode methods				Online-mode methods					
	DECOLOR	RPCA	TVRPCA	MAMR	OMoGMF	ORPCA	incPCP-TI	MEROP	ReProCS	SSMR
Browser1	36.36	33.89	33.96	**37.71**	32.01	36.57	26.04	20.69	20.46	37.46
Bungalows	29.15	28.32	26.54	31.25	33.88	38.13	28.85	26.03	26.03	**39.79**
Cars	39.08	29.44	32.01	**40.42**	34.18	29.73	25.42	23.96	23.95	39.18
Sofa	32.28	32.47	34.37	**39.48**	29.47	34.48	29.07	22.32	26.03	37.08
Office	22.68	20.65	21.56	30.81	21.40	22.47	19.71	18.96	21.19	**35.26**
PETS2006	32.12	25.96	33.05	**43.10**	38.59	34.01	20.80	20.80	23.07	38.43
WinterDriveway	**28.33**	24.94	25.18	28.20	19.86	21.83	23.78	18.82	17.80	27.74
Board	14.20	14.18	14.66	**39.79**	17.35	20.95	13.88	14.16	15.06	37.07
Candela [42]	35.43	34.58	35.71	**44.14**	34.81	31.57	24.86	20.67	21.67	41.89
CAVIAR1	24.98	27.10	35.33	**38.28**	28.39	31.49	20.85	26.04	25.86	36.98
CAVIAR2	26.04	39.89	38.46	**41.13**	39.72	38.50	24.01	25.64	25.64	38.70
CaVignal	35.53	25.06	26.39	**44.89**	23.11	24.14	22.06	20.05	18.67	40.73
Hallmonitor	37.89	29.35	30.35	39.12	**39.85**	35.82	27.13	22.40	23.43	39.14
HighwayI	15.64	38.27	24.00	34.42	34.25	34.00	16.45	26.63	26.64	**39.29**
HighwayII	34.97	36.13	34.01	**36.73**	33.85	34.68	31.29	27.07	24.16	36.62
HumanBody2	31.12	25.64	26.86	**37.31**	23.55	21.93	26.02	18.95	18.64	34.20
IBMtest2	35.42	35.55	32.00	**38.81**	33.32	32.55	28.54	22.43	22.29	36.73
Average	30.07	29.38	29.67	**37.98**	30.45	30.76	24.05	22.09	22.39	37.43

Table 2
SSIM comparison of background recovery with different methods for videos with static background.

	Batch-mode methods				Online-mode methods					
	DECOLOR	RPCA	TVRPCA	MAMR	OMoGMF	ORPCA	incPCP-TI	MEROP	ReProCS	SSMR
Browser1	0.9470	0.9359	0.9103	**0.9523**	0.9337	0.9512	0.9135	0.6519	0.6451	0.9509
Bungalows	0.8343	0.8887	0.8717	0.9464	0.9561	0.9866	0.9558	0.7269	0.7269	**0.9889**
Cars	**0.9877**	0.9596	0.9560	0.9830	0.9670	0.9626	0.9197	0.7895	0.7854	0.9834
Sofa	0.9711	0.9698	0.9719	**0.9876**	0.9629	0.9808	0.9572	0.8093	0.8362	0.9832
Office	0.9358	0.9188	0.9171	0.9823	0.9481	0.9339	0.9052	0.7148	0.7295	**0.9829**
PETS2006	0.9809	0.9572	0.9800	**0.9919**	0.9853	0.9864	0.8913	0.6458	0.6650	0.9873
WinterDriveway	**0.9248**	0.9137	0.9092	0.9180	0.8243	0.8841	0.9114	0.5577	0.5251	0.9152
Board	0.7488	0.6974	0.6965	**0.9806**	0.6282	0.8002	0.6858	0.4354	0.4728	0.9638
Candela	0.9788	0.9805	0.9858	**0.9951**	0.9852	0.9744	0.9331	0.6836	0.7052	0.9939
CAVIAR1	0.9570	0.9526	0.9687	**0.9777**	0.9496	0.9625	0.9071	0.8089	0.8083	0.9723
CAVIAR2	0.9700	0.9847	0.9745	**0.9870**	0.9849	0.9805	0.9541	0.7494	0.7494	0.9845
CaVignal	0.9867	0.9490	0.9382	**0.9937**	0.9190	0.9153	0.9161	0.6228	0.6076	0.9912
Hallmonitor	0.9720	0.9561	0.9399	**0.9741**	0.9739	0.9706	0.9376	0.7426	0.7569	0.9721
HighwayI	0.8954	0.9674	0.8266	**0.9680**	0.9435	0.9410	0.8708	0.8382	0.8370	0.9662
HighwayII	0.9474	0.9479	0.9075	**0.9534**	0.9292	0.9333	0.9289	0.7790	0.7398	0.9527
HumanBody2	0.9834	0.9594	0.9568	**0.9887**	0.9530	0.9069	0.9508	0.6575	0.6496	0.9845
IBMtest2	0.9680	0.9649	0.8975	**0.9734**	0.9595	0.9497	0.9396	0.7253	0.7226	0.9663
Average	0.9405	0.9355	0.9181	**0.9737**	0.9296	0.9424	0.9105	0.7023	0.7037	0.9729

5.3.2. Moving objects detection

With the recovered background, we detect foreground objects through background subtraction. Table 3 reports quantitative results (in terms of the *F*-measure metric) of different methods for foreground detection on some typical video clips from CDnet2014 [55] and SBIdataset [56]. Our method shows the best performance on most of sequences, slightly better than the bacth-mode MAMR [19] by 0.01 on average and outperforming other methods by more than 10%. For more comprehensive comparison, Table 4 shows the average *F-measure* performance of our SSMR model and other competing methods on eight video categories from CDnet2014 dataset [55] including Bad Weather (BW), Baseline (BL), Camera Jitter (CJ), Dynamic Background (DB), Intermittent Object Motion (IOM), Low Framerate (LF), Shadow (SH) and Thermal (TH). The performance of the proposed method shows great robustness over a wide range of video categories and achieves the highest *F-measure* on average, which also verifies the superiority of our SSMR method.

Fig. 5 depicts the visual comparison results of different methods on several sequences. The foreground detection results of our SSMR model are quite close results to the ground truth and those recovered by the batch-mode MAMR [19], DECOLOR [6] and OMoGMF [23] also detect foreground objects accurately for most test clips. However, DECOLOR [6] tends to over-detect the moving objects as observed in the results of *Cars* and *Office*, while OMoGMF [23] suffers from severe foreground leakage from previous frames in *Browser, Office*, and *WinterDriway*. Compared with these three methods, RPCA [2], ORPCA [17], TVRPCA [20], incPCP-TI [42], MEROP [50] and ReProCS [49] cannot handle those challenging cases very well such as *Bungalows*, where almost half area of the frame is occluded by the car. Therefore, it violates the assumption that the foreground is sparse in RPCA-based models, causing a lot of holes in the detected foreground. Moreover, the results of MEROP [50], ReProCS [49], OMoGMF [23], RPCA [2], TVRPCA [20] and MAMR [19] in *PETS2006* show that three persons are standing on the platform, but actually, there are two. That is because the man stands there for a long time and is treated part of the background. Additionally, if the object moves from far to near or from near to far, some pixels would remains constant across a number of frames and are thus separated into the background component. For example in *Office* and *Hallmonitor*, the results of RPCA [2], TVRPCA [20], ORPCA [17] and incPCP-TI [42] appear large areas of holes.

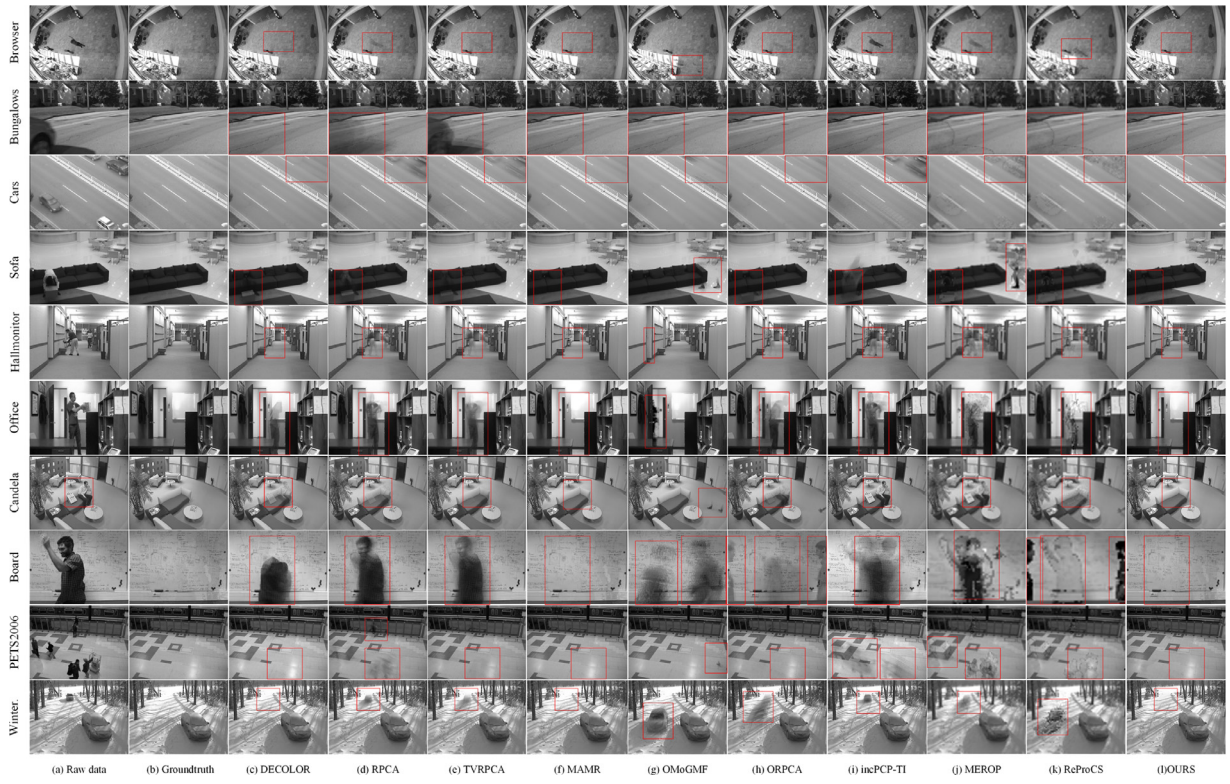

Fig. 4. Visual quality comparison of background recovery for videos with static background on ten clips: (a) Raw data, (b) True background, (c) DECOLOR [6], (d) RPCA [2], (e) TVRPCA [20], (f) MAMR [19], (g) OMoGMF [23], (h) ORPCA [17], (i) incPCP-TI [42], (j) MEROP [50], (k) ReProCS [49], and (l) SSMR (Ours). From top to bottom: background recovery for *Browser1, Bungalows, Cars, Sofa, Hallmonitor, Office, Candela, Board, PETS2006, WinterDriveWay*, respectively.

5.4. Results on videos with dynamic backgrounds

5.4.1. Background recovery

We evaluate the performance of our method on four video sequences with moving backgrounds in CDnet2014 [55], *i.e., Badminton, Boulevard, Sidewalk, Traffic*. As the dataset does not provide ground truth backgrounds, we only present visual comparison on seven state-of-the-art methods handling dynamic backgrounds, *i.e.*, t-MoGMF [23], DECOLOR [6], TVRPCA [20], t-GRASTA [54], incPCP-TI [42], MEROP [50] and ReProCS [49]. For fair comparison, all the algorithms use the same test video clips with blurring frames excluded.

Fig. 6 shows the recovery results of dynamic background. Accurate alignment is crucial to handle dynamic background otherwise the recovered background would be blurred by accumulating non-aligned features. As TVRPCA [20] does not align video frames explicitly, the recovered backgrounds contains significant residue of moving objects at different frames.

The results recovered by t-GRASTA [54], t-MoGMF [23] and incPCP-TI [42] present less amount of foreground leakage due to

Table 3
F-measure comparison of foreground detection with different methods for videos with static background.

	Batch-mode methods				Online-mode methods					
	DECOLOR	RPCA	TVRPCA	MAMR	OMoGMF	ORPCA	incPCP-TI	MEROP	ReProCS	SSMR
Browser1	0.75	0.83	0.80	**0.86**	0.59	0.81	0.59	0.36	0.33	0.84
Bungalow	0.42	0.13	0.14	0.92	0.81	0.91	0.64	0.76	0.81	**0.93**
Cars	0.75	0.71	0.81	0.87	0.75	0.68	0.73	0.66	0.68	**0.90**
Highway	0.91	0.91	**0.96**	0.91	0.87	0.88	0.74	0.84	0.34	**0.96**
Office	0.91	0.57	0.65	**0.93**	0.77	0.72	0.62	0.65	0.86	**0.93**
PeopleInShade	**0.97**	0.93	0.94	0.89	0.88	0.89	0.69	0.93	0.92	0.91
PETS2006	0.86	0.78	0.71	**0.87**	0.79	0.81	0.78	0.68	0.68	**0.87**
WinterDriveway	0.62	**0.63**	0.61	0.55	0.12	0.29	0.57	0.43	0.20	0.55
HallAndMonitor	0.85	0.74	0.80	0.92	0.82	0.83	0.70	0.80	0.79	**0.83**
Board	0.52	0.52	0.46	0.88	0.55	0.68	0.37	0.48	0.69	**0.91**
CAVIAR1	**0.95**	0.83	0.91	0.92	0.79	0.88	0.42	0.71	0.70	**0.95**
CAVIAR2	0.80	0.79	0.75	0.79	0.78	0.80	0.10	0.51	0.48	**0.84**
CaVignal	**0.92**	0.71	0.85	0.88	0.61	0.66	0.86	0.62	0.58	0.83
Candela	0.57	0.43	0.88	0.91	0.47	0.68	0.28	0.58	0.81	**0.93**
HighwayI	0.72	0.64	**0.77**	0.71	0.50	0.41	0.59	0.61	0.61	0.72
HighwayII	0.94	0.94	0.89	0.94	0.60	0.56	0.94	0.73	0.47	**0.95**
HumanBody2	**0.93**	0.62	0.70	0.82	0.43	0.56	0.71	0.62	0.61	0.88
IBMtest2	0.80	0.88	0.80	0.84	0.73	0.75	0.81	0.65	0.66	**0.89**
Average	0.79	0.70	0.75	0.86	0.66	0.71	0.62	0.65	0.62	**0.87**

Table 4
Average *F-measure* comparison on the CDnet2014 dataset. N indicates that those methods cannot handle dynamic videos.

		BW	BL	CJ	DB	IOM	LF	SH	TH	Average
Batch-mode methods	DECOLOR [6]	0.86	0.88	0.66	0.71	0.59	0.79	**0.86**	0.70	0.76
	RPCA [2]	0.85	0.78	N	0.82	0.65	0.78	0.77	0.59	0.75
	MAMR [19]	0.88	0.89	N	**0.86**	**0.77**	0.81	0.85	0.82	0.84
Online-mode methods	OMoGMF [23]	0.75	0.79	0.67	0.76	0.60	0.80	0.66	0.71	0.72
	ORPCA [17]	0.76	0.75	N	0.72	0.62	0.73	0.74	0.78	0.73
	incPCP-TI [42]	0.57	0.71	0.59	0.73	0.61	0.66	0.62	0.59	0.64
	MEROP [50]	0.72	0.65	0.66	0.69	0.50	0.61	0.65	0.75	0.65
	ReProCS [49]	0.56	0.57	0.68	0.67	0.51	0.61	0.60	0.75	0.62
	SSMR	**0.91**	**0.90**	**0.82**	**0.86**	0.75	**0.83**	0.83	**0.88**	**0.85**

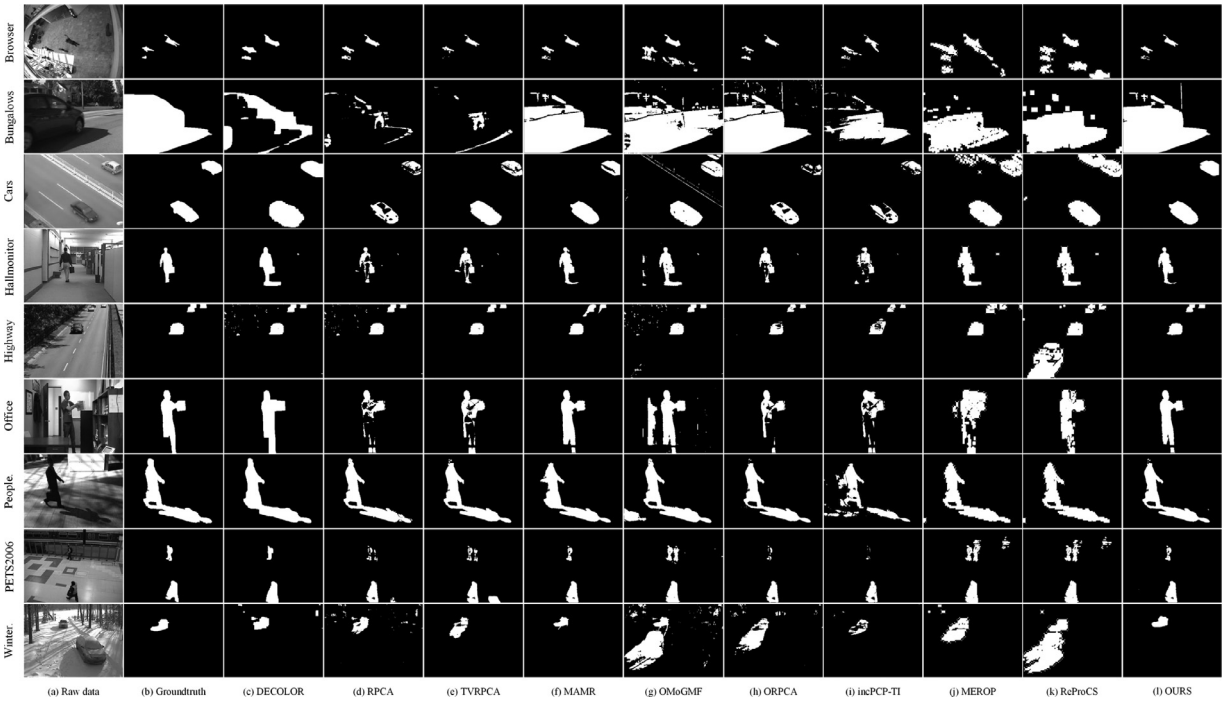

Fig. 5. Visual quality comparison of foreground detection for videos with static background on nine clips: (a) Raw data, (b) Ground-truth, (c) DECOLOR [6], (d) RPCA [2], (e) TVRPCA [20], (f) MAMR [19], (g) OMoGMF [23], (h) ORPCA [17], (i) incPCP-TI [42], (j) MEROP [50], (k) ReProCS [49], and (l) SSMR (Ours). From top to bottom: foreground detection for *Browser1, Bungalows, Cars, Hallmonitor, Highway, Office, PeopleInShade, PETS2006, WinterDriveWay*, respectively.

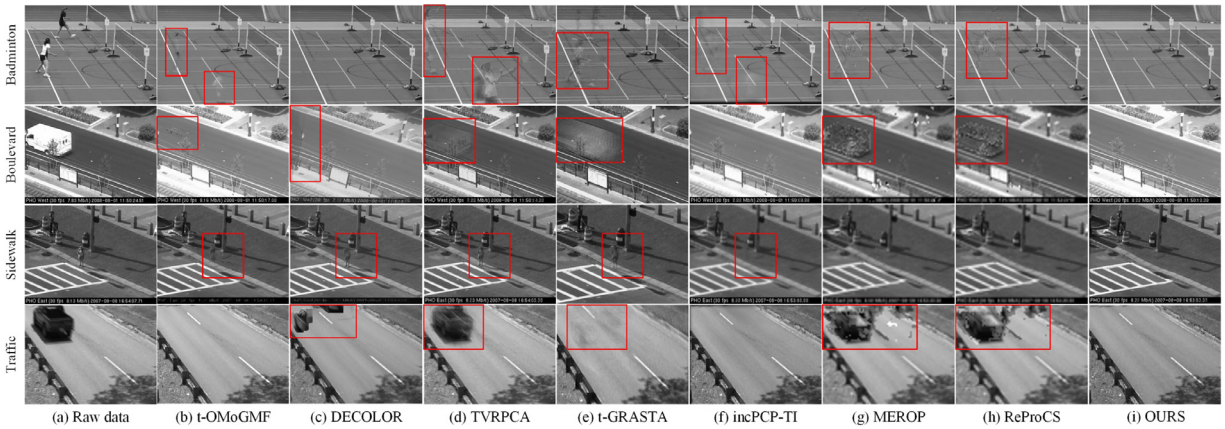

Fig. 6. Visual quality comparison for dynamic videos background recovery on four clips. (a) Raw data, (b) t-OMoGMF [23], (c) DECOLOR [6], (d) TVRPCA [20], (e) t-GRASTA [54], (f) incPCP-TI [42], (g) MEROP [50], (h) ReProCS [49], and (i) RSSMR (Ours). From top to bottom: background recovery for *Badminton, Boulevard, Sidewalk, Traffic*, respectively.

Table 5
F-measure comparison with different methods for dynamic videos foreground detection.

	Badm.	Boul.	Side.	Traf.	Average
t-OMoGMF [23]	0.72	0.81	0.45	0.67	0.67
DECOLOR [6]	**0.86**	0.68	0.49	0.62	0.66
TVRPCA [20]	0.52	0.77	0.08	0.61	0.50
t-GRASTA [54]	0.46	0.62	0.08	0.81	0.49
incPCP-TI [42]	0.55	0.82	0.06	**0.91**	0.59
MEROP [50]	0.55	0.78	**0.77**	0.53	0.66
ReProCS [49]	0.54	0.84	0.76	0.57	0.68
RSSMR	0.73	**0.94**	0.68	**0.91**	**0.82**

effective alignments. The recovered background of *Boulevard* by DECOLOR [6] presents overlapping artifacts as the frequent switching of reference frame suffers from the drifting of alignment errors. In addition, MEROP [50] and ReProCS [49] have similar results, and they both contain severe foreground leakage. Our model recovers clean backgrounds for all the four challenging test sequences thanks to the effective modeling of background motion and efficient numerical algorithm.

5.4.2. Foreground detection

Table 5 shows quantitative results of foreground detection in terms of *F*-measure. Our method achieves the highest average *F*-measure value and outperform the second one by 20%. The visual results in Fig. 7 are also consistent with the quantitative results: foregrounds recovered by our methods are quite close to the ground-truth ones. The foreground detected by DECOLOR [6], t-OMoGMF [23], MEROP [50] and ReProCS [49] include significant amount of background content, while TVRPCA [20] and t-GRASTA [54] tend to yield incomplete detection results, *e.g.*, the persons in *Badminton* and *Sidewalk*. Moreover, our method is able to handle objects with small structures. For example the moving truck in *Boulevard* is occluded by a tree belonging to the background. All the competing methods mis-detect the tree as foreground, while our method accurately separate it from the foreground.

5.5. Results on high-definition videos

In this part, we evaluate our SSMR model on high-definition video clips of size $1920 \times 1080 \times 100$. We randomly select successive 100 frames for validation. Each frame is first decomposed into a two-level pyramid, where the low-frequency subband image of size 480×270 are fed into the matrix recovery model. The recovered background is transformed back to high resolution through foreground masking and pyramid reconstruction. We compared with two competing online methods, OMoGMF [23] and ORPCA [17], as the data volume is to large for batch-mode approaches due to memory limitation. Moreover, the associated results for foreground detection via background subtraction are not presented as they are consistent with the background results. Fig. 8 shows visual quality results of recovered backgrounds from four high-definition video clips, *i.e., Garage1, Garage3, Hall*, and *Campus*. Our method significant outperforms other two online methods for videos with slowly moving objects such as *Garage1, Garage3* and *Hall*, which is also verified by the quantitative results in Table 6. The results demonstrate that the proposed SSMR model not only provide computational-and-space-efficient solution for high-definition videos, but also ensure high performance background modeling.

5.6. Discussions

5.6.1. Computational complexity

For convenient discussion, we restate the key notations: m for number of pixels in a video frame, n the number of frames in a video clip, $m \times r$ the size of background basis $\mathbf{L}$. Our method mainly consists of two parts: motion information extracted by optical flow [51] and alternating minimization of four subproblems: $\mathbf{c}$-subproblem in Eq. (6), $\mathbf{e}$-subproblem in Eq. (6), τ-subproblem in Eq. (18), and $\mathbf{L}$-subproblem in Eq. (11). To solve a $\mathbf{c}$-subproblem, the preparation of the standard least squares form needs $O(mr^2)$ floating point multiplications, and pre-conditioned conjugate gradient algorithm has a computational complexity of $O(r^2\sqrt{\kappa})$, where κ is condition number of the associated normal equations. Therefore, $\mathbf{c}$-subproblems totally needs $O(n(m+\sqrt{\kappa})r^2)$ floating point multiplications. Similarly, the computational complexity of τ-subproblems is $O(nmr)$ as the number of variables of the affine transformation is a constant. With the incremental implementation, the computational complexity of $\mathbf{L}$-subproblems is $O(n(m+r)r^2)$, which is dominated by the matrix inversion and matrix-matrix multiplication in Eq. (15). Each $\mathbf{e}$-subproblem is standard ℓ_1 norm minimization via soft thresholding, and converges for up to three iterations in alternating minimization. Therefore, $\mathbf{e}$-subproblems for n frames totally need $O(nm)$ element-wise shrinkage operations. Moreover, with the pyramidal implementation for high-definition videos, the required computation is reduced to its $1/4^J$, *e.g.*, 1/16 for two-level pyramid decomposition in our implementation.

Our algorithm and other competing ones are implemented in MATLAB, and run on a desktop with Intel i5 2.8 GHz CPU and 8GB RAM. For our algorithm, the percentage of averaged running time for $\mathbf{c}$-subproblems, $\mathbf{e}$-subproblems, τ-subproblems, $\mathbf{L}$-subproblems, and optical flow are 0.15%, 0.02%, 28.01%, 3.20%, and 60.71%, respectively. Table 7 lists statistics of average running times of high resolution videos for single frame. We do not present the results for batch-mode algorithms as they require several hours or cannot even be launched due to memory limitation. The running time

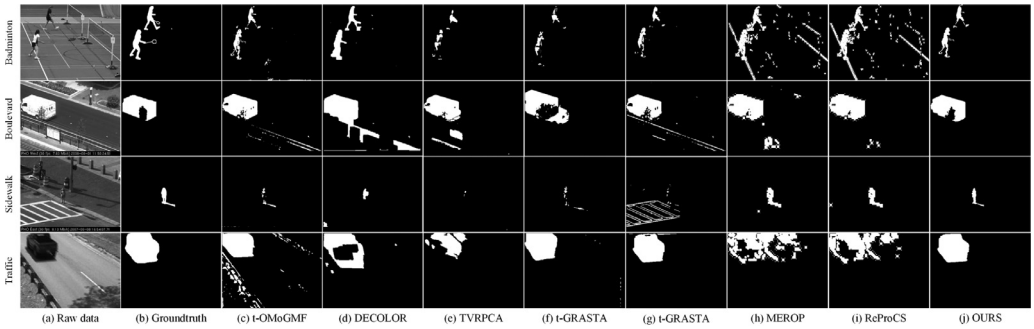

Fig. 7. Visual quality comparison for dynamic videos foreground detection on four clips. (a) Raw data, (b) Groundtruth, (c) t-OMoGMF [23], (d) DECOLOR [6], (e) TVRPCA [20], (f) t-GRASTA [54], (g) incPCP-TI [42], (h) MEROP [50], (i) ReProCS [49], and (j) RSSMR (Ours). From top to bottom: background recovery for *Badminton, Boulevard, Sidewalk, Traffic*, respectively.

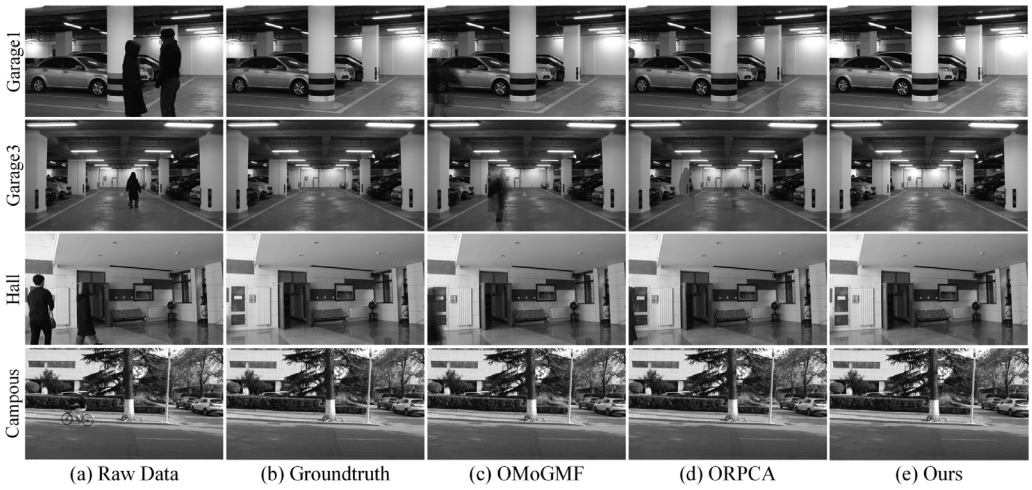

Fig. 8. Visual quality exhibition for high resolution videos on four clips. (a) Raw data. (b) Groundtruth. (c) OMoGMF [23]. (d) ORPCA [17]. (e) SSMR (Ours). From top to bottom: background recovery for *Garage1, Garage3, Hall, Campus*, respectively.

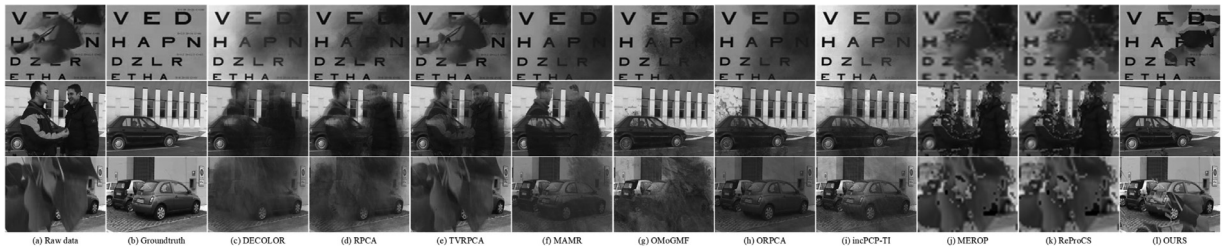

Fig. 9. Failure cases for *Snellen, PeopleAndFoliage* and *Foliage*. (a) Raw data, (b) True background, (c) DECOLOR [6], (d) RPCA [2], (e) TVRPCA [20], (f) MAMR [19], (g) OMoGMF [23], (h) ORPCA [17], (i) incPCP-TI [42], (j) MEROP [50], (k) ReProCS [49], and (l) SSMR (Ours). From top to bottom: background recovery for *Snellen, PeopleAndFoliage* and *Foliage*, respectively.

Table 6
Quantitative comparison for high resolution videos background recovery.

	Garage1		Garage3		Hall		Campus		Average	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
OMoGMF [23]	25.27	0.9603	24.90	0.9688	22.95	0.9715	30.41	0.9667	25.88	0.9668
ORPCA [17]	29.17	0.9696	30.40	0.9745	24.18	0.9699	31.97	0.9628	28.93	0.9692
SSMR	**30.19**	**0.9865**	**32.47**	**0.9845**	**34.39**	**0.9756**	**32.87**	**0.9697**	**32.48**	**0.9791**

Table 7
Average time consuming comparison for high resolution videos.

	Gara1	Gara3	Hall	Camp	Average
OMoGMF [23]	3.29	3.19	3.05	3.08	3.15
ORPCA [17]	5.64	5.90	5.85	5.53	5.73
SSMR	4.77	5.83	5.14	5.60	5.34

of our algorithm is comparable to that of ORPCA [17], and larger than that of t-OMoGMF [23]. As the optical flow module takes up 60.71% of computation, the running time could be reduced using a faster optical flow algorithm. Besides, our algorithm could be further accelerated by more efficient implementation and parallelization.

5.6.2. Failure cases

As shown in Fig. 9, we present some failure cases from SBIdataset [56] for *Snellen, PeopleAndFoliage* and *Foliage*. These clips are extremely complex and the quantitative results of all the competing methods are much lower than other test sequences. All the competing methods suffer from severe foreground leakage due to the violation of sparse prior on the foreground component. For our SSMR model, since most elements in motion weight matrix are zero, the model cannot well recover the background from only few observed pixels. But compared with other methods, our results are visually better. Such cases suggest that, more work is still necessary to challenge these complex scenarios.

6. Conclusions

This paper propose a unified framework called SSMR for background modeling and moving object detection from video sequences. In the proposed model, nuclear norm is used to constrained background components to satisfy the low-rank prior while the foreground is regularized by a sparsity prior. To achieve online implementation, the nuclear norm is formulated as the explicit product of two low-rank matrices. Therefore, the SSMR model is able to process videos frame by frame in the real-time applications. An affine transformation is embedded into our model to automatically compensate camera movements, and is optimized online with other model variables. Moreover, our model is integrated into a multiscale pyramid representation to achieve spatial scalability. An ADM algorithm is derived to solve our SSMR model. Experimental results on various test video sequences demonstrate that our model is able to separate background and moving objects from different types of complex scenes. The proposed model shows

better performance than several batch-mode methods for most cases and outperforms two state-of-the-art online approaches.

Declaration of Competing Interest

None.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61771339, Grant 61571322, Grant 61672378, and Grant 61520106002, and by Tianjin Research Program of Application Foundation and Advanced Technology of China under Grant 18JCYBJC19200 and Grant 17ZXRGGX00160.

References

[1] T. Bouwmans, S. Javed, H. Zhang, Z. Lin, R. Otazo, On the applications of robust PCA in image and video processing, Proc. IEEE 106 (8) (2018) 1427–1457.
[2] E.J. Candes, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM 58 (3) (2011) 1–37.
[3] J. Lei, B. Peng, C. Zhang, X. Mei, X. Cao, X. Fan, X. Li, Shape-preserving object depth control for stereoscopic images, IEEE TCSVT (2017).
[4] W. Dong, G. Shi, X. Hu, Y. Ma, Nonlocal sparse and low-rank regularization for optical flow estimation, IEEE TIP 23 (10) (2014) 4527–4538.
[5] L. Klein, H. Schlunzen, S.K. Von, An advanced motion detection algorithm with video quality analysis for video surveillance systems, IEEE TCSVT 21 (1) (2011) 1–14.
[6] X. Zhou, C. Yang, W. Yu, Moving object detection by detecting contiguous outliers in the low-rank representation, IEEE TPAMI 35 (3) (2013) 597.
[7] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Comput. Surv. 38 (4) (2006) 13.
[8] Y. Tsaig, Automatic segmentation of moving objects in video sequences, IEEE TCSVT 12 (7) (2002) 597–612.
[9] B. Dey, M.K. Kundu, Robust background subtraction for network surveillance in h.264 streaming video, IEEE TCSVT 23 (10) (2013) 1695–1703.
[10] T. Bouwmans, Traditional and recent approaches in background modeling for foreground detection: an overview, Comput. Sci. Rev. 11–12 (2014) 31–66.
[11] L. Maddalena, A. Petrosino, Background subtraction for moving object detection in RGBD data: a survey, J. Imaging 4 (5) (2018).
[12] C.R. Wren, Pfinder: real-time tracking of the human body, IEEE TPAMI 19 (7) (1997) 780–785.
[13] Z. Zivkovic, Improved adaptive gaussian mixture model for background subtraction, in: IEEE ICPR, vol. 2, 2004, pp. 28–31.
[14] N. Bouguila, N. Bouguila, D. Ziou, A robust video foreground segmentation by using generalized gaussian mixture modeling, in: Canadian Conference on Computer and Robot Vision, 2007, pp. 503–509.
[15] T.M. Nguyen, Q.M.J. Wu, Robust student's-t mixture model with spatial constraints and its application in medical image segmentation, IEEE TMI 31 (1) (2011) 103–116.
[16] J. He, L. Balzano, A. Szlam, Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video, in: IEEE CVPR, 2012, pp. 1568–1575.
[17] J. Feng, H. Xu, S. Yan, Online robust PCA via stochastic optimization, in: NIPS, 2013, pp. 404–412.
[18] P. Rodriguez, B. Wohlberg, Incremental principal component pursuit for video background modeling, J. Math. Imaging Vis. 55 (1) (2016) 1–18.
[19] X. Ye, J. Yang, X. Sun, K. Li, C. Hou, Y. Wang, Foreground-background separation from video clips via motion-assisted matrix restoration, IEEE TCSVT 25 (11) (2015) 1721–1734.
[20] X. Cao, L. Yang, X. Guo, Total variation regularized RPCA for irregularly moving object detection under dynamic background, IEEE Trans. Cybern. 46 (4) (2016) 1014–1027.
[21] S. Javed, A. Mahmood, S. Al-Maadeed, T. Bouwmans, S.K. Jung, Moving object detection in complex scene using spatiotemporal structured-sparse RPCA, IEEE TIP 28 (2) (2019) 1007–1022.
[22] T. Bouwmans, A. Sobral, S. Javed, S.K. Jung, E.H. Zahzah, Decomposition into low-rank plus additive matrices for background/foreground separation: a review for a comparative evaluation with a large-scale dataset, Comput. Sci. Rev. 23 (C) (2017) 1–71.
[23] H. Yong, D. Meng, W. Zuo, L. Zhang, Robust online matrix factorization for dynamic background subtraction, IEEE TPAMI PP (99) (2017). 1–1.
[24] J. Yang, J. Yang, X. Yang, H. Yue, Background recovery from video sequences via online motion-assisted RPCA, in: IEEE VCIP, 2017, pp. 1–4.
[25] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, Detecting moving objects, ghosts, and shadows in video streams, IEEE TPAMI 25 (10) (2003) 1337–1342.
[26] C. Stauffer, W.E.L. Grimson, Adaptive background mixture models for real-time tracking, in: IEEE CVPR, 1999, p. 2246.
[27] P. KaewTraKulPong, R. Bowden, An improved adaptive background mixture model for real-time tracking with shadow detection, in: Video-Based Surveillance Systems, 2002, pp. 135–144.
[28] K. Makantasis, A. Nikitakis, A.D. Doulamis, N.D. Doulamis, I. Papaefstathiou, Data-driven background subtraction algorithm for in-camera acceleration in thermal imagery, IEEE TCSVT 28 (9) (2018) 2090–2104.
[29] R.H. Evangelio, M. Patzold, I. Keller, T. Sikora, Adaptively splitted GMM with feedback improvement for the task of background subtraction, IEEE TIFS 9 (5) (2014) 863–874.
[30] T.S. Haines, T. Xiang, Background subtraction with dirichlet processes, in: ECCV, Springer, 2012, pp. 99–113.
[31] H.-C. Wang, Y.-C. Lai, W.-H. Cheng, C.-Y. Cheng, K.-L. Hua, Background extraction based on joint gaussian conditional random fields, IEEE TCSVT 28 (11) (2017) 3127–3140.
[32] A. Elgammal, R. Duraiswami, D. Harwood, L.S. Davis, Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, Proc. IEEE 90 (7) (2002) 1151–1163.
[33] Z. Gao, L.F. Cheong, M. Shan, Block-sparse RPCA for consistent foreground detection, in: ECCV, 2012, pp. 690–703.
[34] A. Sobral, T. Bouwmans, E.H. Zahzah, Double-constrained RPCA based on saliency maps for foreground detection in automated maritime surveillance, in: IEEE International Conference on Advanced Video and Signal Based Surveillance, 2015.
[35] S. Javed, A. Mahmood, T. Bouwmans, S.K. Jung, Spatiotemporal low-rank modeling for complex scene background initialization, IEEE Trans. Circuits Syst. Video Technol. PP (99) (2016). 1–1.
[36] S. Javed, T. Bouwmans, M. Sultana, S.K. Jung, Moving object detection on RGB-D videos using graph regularized spatiotemporal RPCA, in: International Conference on Image Analysis and Processing, 2017.
[37] K. Taehyeon, C. Yoonsik, Real-time background subtraction via l1 norm tensor decomposition, in: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2018.
[38] L. Li, P. Wang, Q. Hu, S. Cai, Efficient background modeling based on sparse representation and outlier iterative removal, IEEE TCSVT 26 (2) (2016) 278–289.
[39] N. Vaswani, T. Bouwmans, S. Javed, P. Narayanamurthy, Robust subspace learning: robust PCA, robust subspace tracking and robust subspace recovery, IEEE Signal Process. Mag. 35 (4) (2018) 32–55.
[40] Y. Peng, A. Ganesh, J. Wright, W. Xu, Y. Ma, RASL: robust alignment by sparse and low-rank decomposition for linearly correlated images, IEEE TPAMI 34 (11) (2012) 2233–2246.
[41] S.E. Ebadi, V.G. Ones, E. Izquierdo, Efficient background subtraction with low-rank and sparse matrix decomposition, in: IEEE ICIP, 2015, pp. 4863–4867.
[42] P. Rodriguez, B. Wohlberg, Translational and rotational jitter invariant incremental principal component pursuit for video background modeling, in: IEEE ICIP, 2015, pp. 1844–1852.
[43] J. He, L. Balzano, A. Szlam, Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video, in: IEEE CVPR, 2012, pp. 1568–1575, doi:10.1109/CVPR.2012.6247848.
[44] J. Xu, V.K. Ithapu, L. Mukherjee, J.M. Rehg, V. Singh, GOSUS: Grassmannian online subspace updates with structured-sparsity, in: IEEE ICCV, 2013, pp. 3376–3383.
[45] N. Wang, T. Yao, J. Wang, D.-Y. Yeung, A probabilistic approach to robust matrix factorization, in: ECCV, 2012, pp. 126–139.
[46] S. Javed, A. Sobral, S.H. Oh, T. Bouwmans, S.K. Jung, Background subtraction via superpixel-based online matrix decomposition with structured foreground constraints, in: IEEE International Conference on Computer Vision Workshop, 2016.
[47] C. Qiu, N. Vaswani, Recursive sparse recovery in large but correlated noise, Communication, Control, and Computing, 2011.
[48] G. Han, C. Qiu, N. Vaswani, Practical ReProCS for separating sparse and low-dimensional signal sequences from their sum, Signal and Information Processing, 2015.
[49] P. Narayanamurthy, N. Vaswani, Provable dynamic robust PCA or robust subspace tracking, IEEE TIT 65 (3) (2019) 1547–1577.
[50] P. Narayanamurthy, N. Vaswani, A fast and memory-efficient algorithm for robust PCA (MEROP) (2018) 4684–4688.
[51] T. Brox, J. Malik, Large displacement optical flow: descriptor matching in variational motion estimation, IEEE TPAMI 33 (3) (2011) 500–513.
[52] M.N. Do, M. Vetterli, Framing pyramids, IEEE TSP 51 (9) (2003) 2329–2342.
[53] S. Richard, Computer vision: algorithms and applications, 2010.
[54] J. He, D. Zhang, L. Balzano, T. Tao, Iterative Grassmannian optimization for robust image alignment, Image Vis. Comput. 32 (10) (2013) 800–813.
[55] W. Yi, P.M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, P. Ishwar, CDnet 2014: An expanded change detection benchmark dataset, Computer Vision and Pattern Recognition Workshops, 2014.
[56] L. Maddalena, A. Petrosino, Towards benchmarking scene background initialization, in: ICIAP, 2015, pp. 469–476.