Pattern Recognition 100 (2020) 107153

Contents lists available at ScienceDirect

Pattern Recognition

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

Statistical bootstrap-based principal mode component analysis for dynamic background subtraction

Benson S.Y. Lam [a,*], Amanda M.Y. Chu [b], H. Yan [c]

[a] Department of Mathematics and Statistics, The Hang Seng University of Hong Kong, Shatin, Hong Kong, China
[b] Department of Social Sciences, The Education University of Hong Kong, Ting Kok, Hong Kong, China
[c] Department of Electrical Engineering, City University of Hong Kong, Kowloon Tong, Hong Kong, China

ARTICLE INFO

Article history:
Received 20 June 2019
Revised 2 October 2019
Accepted 8 December 2019

Keywords:
Background modeling
Video surveillance
Principal Component analysis
Statistical mode

ABSTRACT

Background subtraction is needed to extract foreground information from a video sequence for further processing in many applications, such as surveillance tracking. However, due to the presence of a dynamic background and noise, extracting foreground accurately from a video sequence remains challenging. A novel projection method, namely Principal Mode Component Analysis (PMCA), is proposed to capture the most repetitive patterns of a video sequence, which is one of the key characteristics of the video background. The patterns are captured by applying the bootstrapping method together with the statistic mode measure. The bootstrapping method can model the distribution of almost any statistic of the dynamic background and complicated noise. This is different from current methods, which restrict the distribution to a closed-form function. We introduce a mathematical relaxation that can formulate the statistical mode measure for a continuous video data. A fast exhaustive search method is proposed to find the global optimal solution for the PMCA. This fast method adopts a simplification procedure that makes the optimization procedure independent of the video size. The proposed method is computationally much more traceable than existing ones. We compare the proposed method with 10 different methods, including several state-of-the-art techniques, for 19 different real-world video sequences from two popular datasets. Experiment results show that the proposed method performs the best in 16 cases and second best in 2 cases.

1. Introduction

Background subtraction is an essential and fundamental step in various video-processing problems, such as moving object detection [1], object tracking [2], and video surveillance. Due to its importance, many different background subtraction methodologies have been proposed in recent years [3–5]. One of the common objectives of applying a background subtraction methodology and analyzing a video sequence is to separate a moving foreground from a background. An example is a traffic camera that records information about vehicles moving on a road. The foreground comprises the moving vehicles while the background comprises the road.

The major challenge of the background subtraction problem is that a real-world video background is not strictly static but usually contains noise and dynamic elements such as lighting changes, wavering tree branches, and fountain waterfalls. It is hard to distinguish moving foreground objects from dynamic backgrounds. One example is the video surveillance system that monitors the road situation [6]. The system of interest may be the moving buses and/or moving cars. However, the roadside may have wavering tree branches. As these tree branches are in motion, which is similar to the moving buses/cars, this can confuse current methods and incorrectly classify the branches as part of the foreground.

Many different methods have been proposed to solve the background subtraction problem. However, these techniques still have evident defects when applied to real-world videos. The first methods used to estimate a video background are based on the median, mean, and histogram of all of the pixels across a number of video frames [7]. After the background is obtained, the foreground is obtained by examining the difference between the obtained background and each video frame. Pixels exhibiting large differences across each video frame imply a high possibility of foreground objects. However, this approach is mainly designed for video sequences with static backgrounds. Dynamic backgrounds such as wavering tree branches may be incorrectly classified as foregrounds. To address these problems, numerous solutions have been proposed [3–5]. These solutions are mainly based on a

* Coressponding author.
E-mail address: bensonlam@hsu.edu.hk (B.S.Y. Lam).

https://doi.org/10.1016/j.patcog.2019.107153
0031-3203/

method known as robust principal component analysis (RPCA) [8], which decomposes a video sequence into a single low-rank component and a sparse component. The optimization problem is given as follows:

$$\min_{\mathbf{L},\mathbf{E}} \mathbf{L}_* + \lambda \mathbf{E}_1, \ \text{s.t.}\ \mathbf{Y} = \mathbf{L} + \mathbf{E}$$

where $\mathbf{Y} \in R^{m_l m_w \times t}$ is the input video sequence and $m_l m_w$ and t are the video size and number of frames in the video sequence, respectively. λ is a user-defined parameter. $\mathbf{X}_*$ and $\mathbf{X}_1$ are the nuclear and l_1 norms of the matrix $\mathbf{X}$, respectively. The solution matrices $\mathbf{L}$ and $\mathbf{E}$ are the low-rank and sparse matrices, respectively. The low-rank matrix approximates the background variations such as wavering tree branches while the sparse matrix models the foreground objects. Although this method gives more promising results than the classical background subtraction method, it implicitly assumes that the sparse error, or matrix $\mathbf{E}$, follows a Laplace distribution, which may not be the case in real-world video sequences. Different attempts have been made to replace the Laplace assumption with different distribution functions, such as a mixture of Gaussian distributions [9], a mixture of Power Laplace distribution [10], and the Dirichlet process [11]. Although these methods can produce more accurate video backgrounds, it is difficult to accurately represent real-world dynamic backgrounds and general noise structures with a closed-form distribution function. Moreover, the introduction of a mixture distribution usually involves hyperparameters such as the number of components and the noise level. This hinders the applicability of these methods to real-world video problems. Another popular approach to handling the dynamic background problem is to impose total variation regularization on the sparse error [12–14]. This requires the foreground objects to be continuous and to change slowly throughout the video time frame. Although this approach can suppress the negative effects of dynamic backgrounds, it introduces user-defined parameters that are usually data-dependent. Moreover, the regularization is embedded in the optimization procedure, and tuning the parameters needed to solve the optimization problem can be very time-consuming.

To alleviate the aforementioned issues, we propose a novel methodology that is completely different from the distributional and total variation approaches. The key idea of the proposed method is to model the dynamic background and complicated noise via statistical bootstrapping and statistical mode formulation. We first apply statistical bootstrapping to the video sequence and obtain a set of subsamples (Section 3.1). In Section 3.2, we modify the l_1-PCA model and propose a novel PCA method known as principal mode component analysis (PMCA) to obtain the statistical mode of each subsample. The statistical mode is the statistic that captures the most repetitive pattern of the subsamples. By combining all of these mostly repetitive patterns, a powerful method that captures the dynamic background and complicated noise is introduced. By examining the differences between these modes and the original video frames, we obtain an outlier map that contains preliminary information about the foreground objects. The final foreground objects are obtained by applying a morphological operation to the outlier map. Fig. 1 shows a flowchart for this whole procedure.

Our work makes the following contributions.

- We propose the use of the bootstrapping technique for solving background subtraction problems. The bootstrapping technique has proved to be an effective tool for estimating almost any statistic using random sampling methods. This statistical method has been widely applied in various areas [15,16]. In contrast to the distributional methods, the proposed bootstrapping method can represent a broader range of dynamic structures without knowing the closed form distribution function.
- We introduce a novel projection method, namely, PMCA, to capture the repetitive patterns of a set of sampled video sequences. We use this novel method to find the statistical mode of the selected samples. As the statistical mode is well defined only for categorical data, we propose a relaxed version of the statistical mode that is applicable to video data. We also apply a fast exhaustive search algorithm to this problem and obtain a global optimal solution.
- The proposed method has a much lower computational complexity than most background subtraction methods. We propose a robust optimization scheme that can quickly identify the statistical mode measure of a bootstrapped sample. This optimization scheme is not dependent on the video size after some simplifications. This makes the proposed method much more computationally traceable than current methods that must apply a set of procedures to the whole dataset many times to attain convergence.
- We adopt a flexible regularization method and allow the parameter-tuning procedure to be tunable without solving the whole mathematical problem again.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of the related work on RPCA-based background subtraction methods. We then introduce our proposed bootstrap-based PMCA in Section 3. After that, we demonstrate the robustness of the proposed method via various experiments in Section 4. Section 5 presents our conclusions and possible future work.

2. Related work

In this section, we briefly review the background subtraction methods that are related to the RPCA. These methods can be broadly classified as the probabilistic, spatial-temporal, and online learning approaches. We also review the bootstrapping method that is nowadays a widely used technique in various pattern recognition problems.

2.1. Probabilistic approach

The probabilistic approach is adopted to impose prior probability to each of the decomposed components of the RPCA model. Wang *et al.* [17] proposed a probabilistic robust matrix factorization that formulated the sparse component with a Laplace prior and the two matrices for the low-rank component with a Gaussian prior. However, these assumptions may be too restrictive and limit the model's ability to handle real-world situations. In addition, this method models the foreground objects in a pixelwise manner. This may contradict the characteristics of real-world foreground objects in that a foreground usually forms groups with a high within-group spatial proximity. Later, Wang *et al.* [18] improved the probabilistic robust matrix factorization model and proposed a new framework based on Bayesian formulation. They imposed conjugate priors (multivariate normal distribution and Wishart distribution) onto the low-rank component and a generalized inverse Gaussian distribution onto the sparse errors. They also introduced a Markov extension to the model that introduced the within-group spatial proximity effect, which linked the connections between different object pixels. Although this imposition offers a higher flexibility for model noise and enhances the model's robustness, it assumes that the video noise follows a single distribution, which may limit its applicability to real-world data with complex noise. To deal with general types of noise, several different works have attempted to make the model more adaptable by making different probabilistic assumptions about the noise in video signals. Zhao *et al.* [19] modified the aforementioned Bayesian framework and introduced a two-level generative Gaussian approach to model the

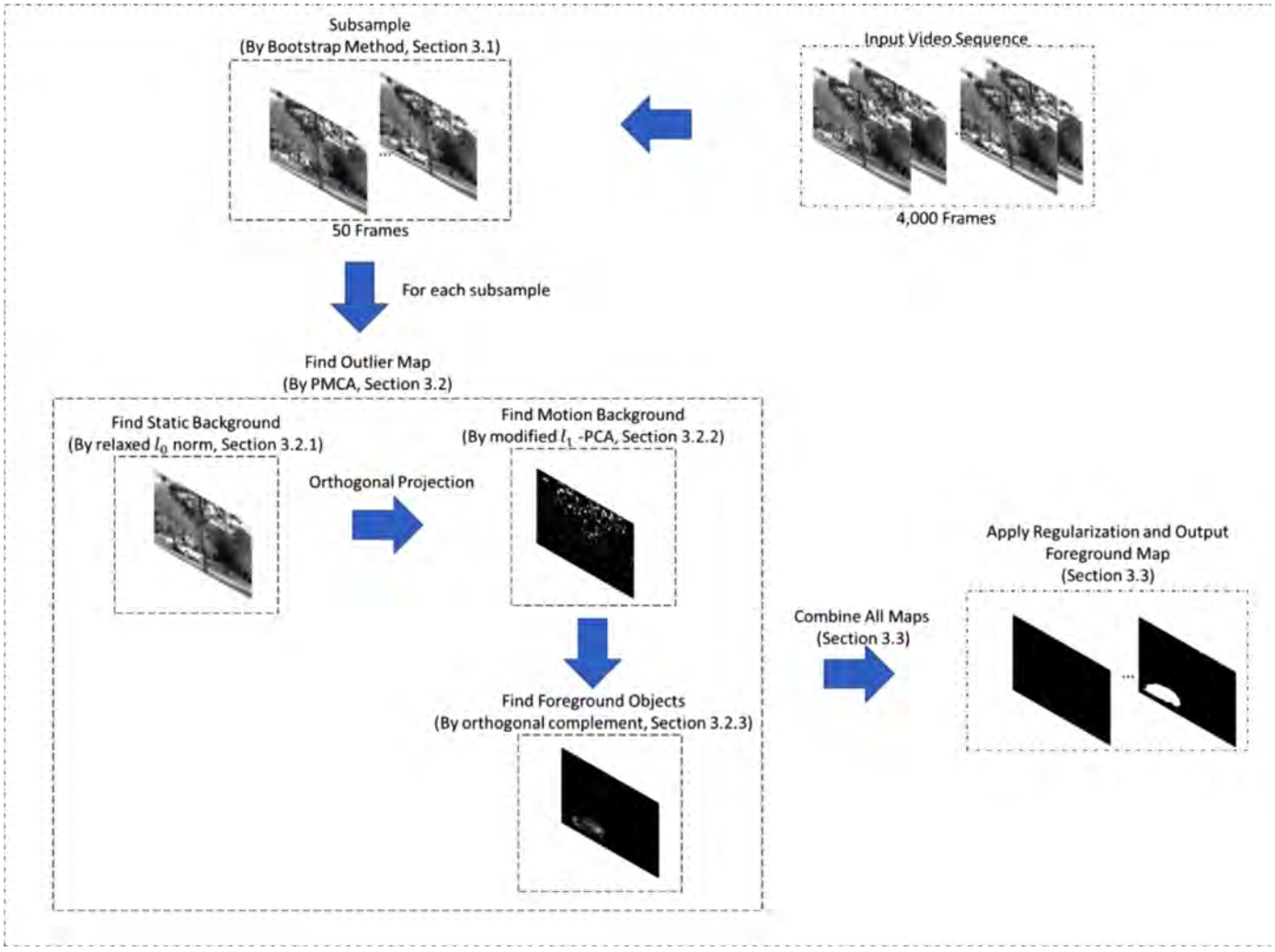

Fig. 1. The procedure of the proposed method.

complex noise. Meng *et al.* [9] modified the two-level generative approach and adopted a mixture of Gaussian distribution to model a more general noise. This work was extended to a non-parametric Bayesian adaptive approach by Chen *et al.* [11], who incorporated the Dirichlet process into the aforementioned mixture of Gaussian distribution. Cao *et al.* [10] assumed that the complex noise followed a mixture of exponential power distributions. Each component in the proposed mixture model is adapted from a series of preliminary super or sub-Gaussian distributions. Lakshminarayanan *et al.* [20] proposed a Gaussian scale mixture model that could model noise with a heteroscedastic structure. Babacan *et al.* [21] introduced a different approach and decomposed the sparse errors into two different matrices: the sparse and dense error matrices, which are governed by two different Gaussian distributions. A concept that is similar to the Gaussian mixture model called kernel density estimation (KDE) is also adopted for background modeling. Zhang *et al.* [22] proposed kernel density estimation (KDE) to model the video background and adopted the idea that is similar to the computer game Tetris to update the background iteratively. Berjon *et al.* [23] modeled both the foreground and background of the video sequence using the KDE and proposed the use of particle filter to track the motion of the foreground. Although these probabilistic methods can produce very good results, Wang *et al.* [24] argued that the assumptions were too restrictive, requiring the noise of a real-world video to follow a single distribution or a mixture of distributions and limiting the applicability of these methods in solving practical video-processing problems. Moreover, the introduction of hyperparameters such as the number of noise distributions, noise variation, and noise level largely increases model complexity and redundancy. This implies that the probabilistic approach cannot fully reveal a complex noise structure and that there remains a need to model the complex noise more effectively. In fact, Wang *et al.* [24] suggested adopting a non-probabilistic approach to capture the key characteristic of the noise signals. They applied the Fourier transform to the video signals and transformed them in the frequency domain before applying a modified RPCA model to the transformed data. However, this method has been shown to perform well only for synthetic data and four examples of real-world video sequences.

2.2. Spatial-temporal approach

Another type of background subtraction technique is to impose spatial-temporal constraints to suppress the negative effects of dynamic backgrounds and to obtain more precise foreground objects. In many real-world video-processing problems, the dynamic background and complex noise signal can corrupt the shapes of foreground objects and obtain objects with missing regions. Various regularization techniques including non-negative matrix factorization [25], l_p norm regularization [26], and total variation [27] have been widely applied to introduce additional information and fill in the missing regions of the detected foreground objects. Guo *et*

al. [28] incorporated total variation regularization into the sparse error term of the RPCA model. This imposes a smoothness constraint on the detected foreground objects. Zhou *et al.* [1] proposed a model for detecting contiguous outliers in the low-rank representation (DECOLOR), which decomposes a video input into the foreground and background components. To obtain smooth foreground objects, total variation regularization is applied to the foreground modeling. Cao *et al.* [13] modified the RPCA model by further decomposing its sparse error term into two different parts: the intrinsic foreground and dynamic background. The intrinsic foreground is obtained by constraining the output to be smooth, and total variation regularization is applied. The dynamic background is formulated as another sparse error, and another l_1 norm minimization is introduced. Later, Xue *et al.* [12] modified this method by introducing a rank-1 constraint to the model. This rank-1 constraint can model the linear change across video frames and provides a more effective way to capture the dynamic component of the video data. Cao *et al.* [14] introduced the concept of tensor decomposition to handle the background modeling problem. The video input is represented not by a two-dimensional matrix but by a three-dimensional matrix known as the tensor. The authors decompose the video signal into three different parts, including a low-rank matrix, sparse noise, and a foreground, which is a continuous tensor. The continuity of the three-dimensional tensor is formulated by applying a three-dimensional version of total variation regularization. Similar to Cao *et al.* [14], Xia *et al.* [17] proposed a multidimensional tensor and introduced three-dimensional total variation regularization. Woo and Park [29] proposed a new type of model that decomposed a video signal into a low-rank non-negative matrix and a l_p norm sparse error with a total variation regularization constraint. In a low-rank non-negative formulation requires a small number of non-negative bases in the detected background and assumes the background is slowly varying. In contrast to the low-rank formulation of the RPCA model, this non-negative matrix factorization has an inherent clustering property that automatically clusters the frames of the video input [30]. This imposes spatial correlations on the background information. The l_p norm, together with the total variation regularization constraint, requires the foreground to be sparse and the objects of the foreground to be continuous. Yu *et al.* [31] combined different l_p norms with deep learning methodology [32–34] to extract the foreground from an extremely low-resolution video sequence. Although the method works very well, it introduces several different user-defined parameters to indicate the contributions of different l_p norms. If any parameter setting is changed, the whole problem has to be solved again, which may be very time-consuming. Although these methods can preserve the shape of detected foreground objects better and suppress the negative effect of dynamic backgrounds more effectively, the spatial-temporal constraint is usually embedded in the optimization procedure. Tuning the parameters requires solving the whole optimization problem again, which is very time-consuming.

2.3. Online learning approach

Recently, some works have been devoted to the study of online subspace learning methods to handle real-time background modeling problems [35]. The core idea is to incrementally compute only one frame at a time and gradually enhance the background estimation by updating each frame. The state-of-the-art approaches along this research line mainly include OPRMF [17], GRASTA [36], GOSUS [37], PracReProCS [38], MeDRoP [39], and incPCP [40,41]. GRASTA uses an L_1 norm loss for each frame to encode sparse foreground objects and applies the ADMM technique to update the subspace. Similar to GRASTA, OPRMF adopts the L_1 norm formulation and imposes an additional regularization term onto subspace parameters to avoid overfitting. GOSUS proposes the use of group sparsity to characterize the structure of a video foreground. In addition, the recently proposed PracReProCS and incPCP are the incremental extensions of the classical PCP algorithm. Although these methods work very well for the real-time background problems, most of them rely on the incremental update that computes only one frame at a time. This is very time-consuming if the video sequence is very long.

2.4. Bootstrapping

Bootstrapping method has been widely used in various pattern recognition problems including object classification [42], handwritten digit recognition [43], mode seeking [44], etc. One of the most well-known bootstrapping-based classifiers is the Random Forest. Its essential idea is to perform sampling to the original dataset and then apply the Decision Tree method to each of the subsample. After that, they adopt majority principle to determine the class labels of the objects. A major merit of the bootstrapping method is that it can model nearly any distribution. It is also proved to be robust to handle various complicated problems. Although this method has been applied in various areas, there is not much change about sampling procedure. Even to the three example applications (object classification [42], handwritten digit recognition [43], mode seeking [44]), the authors apply the bootstrapping method by sampling the original data and then apply the proposed method to the subsamples. However, to the best of our knowledge, it is hard to find research work that applies bootstrapping method to solve background modeling problems.

In this paper, we propose a novel bootstrapping-based projection method called principal mode component analysis (PMCA). It can address the major limitations of the existing background subtraction methods. The probabilistic approach assumes the distributions to have closed-form expressions, which confines the applications of existing methods to more general situations. The proposed method adopts bootstrapping method, which can model nearly any distribution and thus is able to represent a boarder range of dynamic structure without knowing the closed-form distribution function. The spatial-temporal approach introduces regularizations to better preserve the shapes of the extracted foregrounds. However, tuning the parameters need to solve the whole optimization problem again, which can be very time-consuming. The proposed method first adopts a fast method to find the global optimal solution of the problem, which takes only several seconds to extract the foreground from a video sequence with over 4000 frames. Moreover, we introduce the use of morphological operation that does not need to solve the optimization problem again. For the online learning approach, it adopts an incremental update that computes only one frame at a time, which can be time-consuming. The proposed method adopts a fast approach to solve the optimization problem.

3. Methodology

The flowchart of the proposed method is shown in Fig. 1. In Section 3.1, the statistical bootstrap method generates a set of subsamples. For each subsample, we perform a series of operations (Section 3.2). First, a relaxed l_0 norm measure is used to find the most repetitive static background (Section 3.2.1). Then, an orthogonal projection is performed to the subsampled sequence, which can eliminate the static background of the subsampled sequence. The motion background (such as wavering tree branches) are extracted by a combination of the relaxed l_0 norm measure and l_1-PCA method (Section 3.2.2). After that, the foreground objects of the subsample can be identified by taking orthogonal complement of the static and motion backgrounds. Finally, by combining all the

Table 1
The procedure for statistical bootstrapping.

Input: Original dataset **X**, the subsample size m
Output: Distribution of the statistic **u**
Step 1. $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ are subsamples that are drawn with replacement from the original dataset **X**.
Step 2: Compute a statistic **u** from the drawn subsamples.
Step 3: Repeat Steps 1 and 2 many times.

foreground objects from all subsamples and applying the regularization (Section 3.3), we obtain the foreground map.

3.1. Statistical bootstrapping for modeling the distribution of a video background

In statistics, the bootstrapping method is used to apply random sampling with replacement to the collected samples, also known as a subsample, and to estimate the distribution of a statistic, such as an arithmetic mean [15,16]. This method has proved to be powerful and can model almost any statistic. It has been widely applied to various image processing, computer vision, and pattern recognition problems. The bootstrapping method procedure is briefly summarized in Table 1. The sample size n_{boot} controls the number of possibilities included in the calculation of the statistical mode measure. In our study, this parameter controls the number of video background frames included in the study. If the repetitive patterns dominate the video sequence, a small n_{boot} is enough to capture the video background. However, if the repetitive patterns do not dominate the video sequence, a larger n_{boot} may be needed. In our experiments, the repetitive patterns dominate the video sequences and a small sample size $n_{boot} = 50$ gives good results.

3.2. Modeling the statistic of a dynamic background by principal mode component analysis (PMCA)

To model the statistic of a dynamic background, we propose a novel PCA method, namely, PMCA, that decomposes a subsample into a set of orthonormal projection vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$. That is, for a given video subsample $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_m]$,

$$\mathbf{X} = \alpha_1 \mathbf{v}_1 \mathbf{v}_1^T + \ldots \alpha_m \mathbf{v}_m \mathbf{v}_m^T,$$

where $\alpha_1, \ldots, \alpha_m$ are the projection coefficients. We require that the first several projection vectors extract the most repetitive patterns of the video sequence. In many real-world video sequences, dynamic backgrounds are usually periodic and exhibit repetitive patterns, such as the wavering tree branches. These most repetitive patterns are the items that appear most throughout the video sequence and thus can be modelled by the statistical mode measures.

3.2.1. Relaxed $l_0$norm expression

To capture the most repetitive patterns of a video sequence, we adopt the l_0 norm, the sparse version of which has been widely used in numerous computer vision and pattern recognition problems [8,45]. The l_0 norm counts the number of non-zero elements in a vector [46]. Given a categorical vector $\mathbf{x} = [x_1, x_2, \ldots x_m]^T$, the statistical mode can be obtained by minimizing the following optimization problem:

$$m_q = \underset{c}{\operatorname{argmin}} \mathbf{x} - c1_0, \tag{1}$$

where 1 is a vector of ones. The global optimal solution m_q is the element that can cause the most categorial elements of the vector to be zero. In other words, this is the statistical mode of the data. However, the l_0 norm does not have a closed-form mathematical expression, which makes the optimization problem difficult and not generalizable to quantitative data. We apply a theory from classical mathematics to relax the l_0 norm and express it in a continuous function. By taking the limit p of the l_p norm to be zero, the l_p norm approaches the l_0 norm. Mathematically, this is $\lim_{p\to 0} f_p = \exp\{\int_X \log|f| d\mu\}$ [46]. Its direct discrete approximation is then given by $\lim_{p\to 0} \mathbf{x}_p = \exp\{\sum_i \log|x_i|\}$. With this relaxed l_0 norm expression, the optimization problem in Eq. (1) becomes

$$\min_c \exp\left(\sum_{i=1}^{m} \log|x_i - c|\right). \tag{2}$$

If $\mathbf{x} = [x_1, x_2, \ldots x_m]^T$ is a categorical vector, the global optimal solution of Equation (2) is the statistical mode of **x**. When c is the most frequently appearing item, more logarithm terms of the objective function become negative infinity, and thus a smaller value is obtained. To avoid singularity in Equation (2), we regularize the objective function as $\min_c \exp(\sum_i \log(\epsilon + |x_i - c|))$, where ϵ is a small value taken as 10^{-4} in our experiments. As the exponential function is monotonic, the optimization problem can be simplified as follows:

$$\min_{\mathbf{c}} \sum_{i=1}^{m} \log\left(\epsilon + \mathbf{x}_i - \mathbf{c}_2^2\right). \tag{3}$$

Here, we apply the l_2 norm to generalize the optimization problem and apply it to multivariate and quantitative data. The solution vector **c** is the relaxed version of the statistical mode of the subsample. By projecting the subsamples to this vector space, the mostly repetitive pattern of the video samples is obtained. The vector space is constructed by normalizing the vector of the solution of Eq. (3), that is, $\mathbf{v}_1 = \mathbf{c}/\mathbf{c}_2$. In background modeling, the most repetitive pattern of a subsample is the static background. Therefore, the vector space spanned by v_1 captures the static background of the video sequence. To find the motion repetitive patterns, we construct more projection vectors and we apply an idea that is similar to the widely used l_1-PCA method. This is explained in next section.

3.2.2. Constructing the rest of the projection vectors

To capture the most repetitive patterns of the motion background, we construct p^{th} projection vectors with each of which capture some mostly repetitive motion patterns. This can be achieved by applying a modified version of the widely used l_1-PCA model [47]:

$$\min_{\mathbf{c}, s_i \in \{-1,1\}} \sum_{i=1}^{m} d(\mathbf{x}_i, s_i\mathbf{c}), \text{ where } d(\mathbf{x}_i, s_i\mathbf{c}) = \log\left(\epsilon + \mathbf{x}_i - s_i\mathbf{c}_2^2\right). \tag{4}$$

This optimization problem is similar to Eq. (3) except that the binary variable $s_i \in \{-1, 1\}$ is introduced. The l_1-PCA model has proved to be a robust tool for extracting the most useful features of data. This model adopts the Euclidean distance, which takes $d(\mathbf{x}_i, s_i\mathbf{c}) = \mathbf{x}_i - s_i\mathbf{c}_2^2$ in Eq. (4). This implicitly partitions the data into two clusters, and vector **c** is the direction vector between the two cluster representatives that are computed as the signed mean of the data. This is mathematically justified, and the details can be

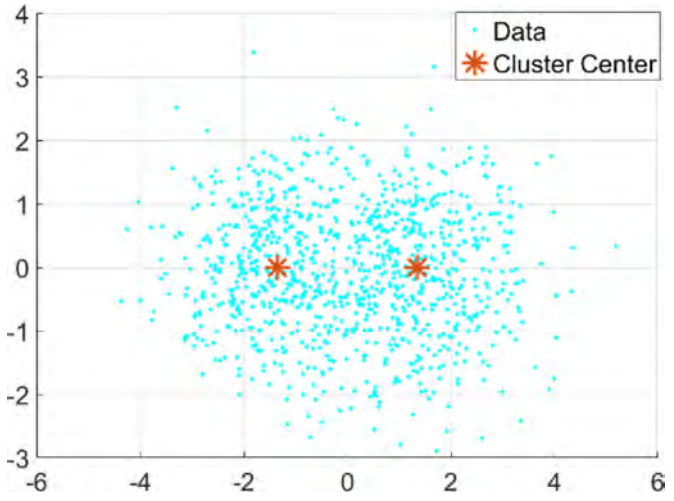

Fig. 2. Illustration of the l_1-PCA.

found in Appendix A (in the online supplementary materials). This property of the l_1-PCA model is illustrated in Fig. 2, which shows two spherical clusters. $s_i = -1$ and $s_i = 1$ can be treated as labels of the data samples. The samples of the left cluster are represented by $s_i = -1$ while the samples of the right cluster are represented by $s_i = 1$. Vector $\mathbf{c}$ is the direction vector between the two cluster centers, which are the means of the two clusters. However, for the background subtraction problem, the means adopted by the l_1-PCA model are not robust to outliers, which are the foreground objects of the video sequence and always exist in a video sequence. The proposed model (Eq. (4)) uses a logarithm of Euclidean distance, which is a robust to the presence of outliers. Moreover, due to the characteristics of the statistical mode, the vector $\mathbf{c}$ projects the data to their mostly repetitive pattern. Similar to the first component, the projection vector is obtained by normalizing the solution vector with the l_2 norm.

We introduce fast exhaustive search methods to find the global optimal solutions to both the statistical mode problem and the modified l_1-PCA problem. Intuitively, the global optimal solutions of Equations (3) and (4) are one of the samples of the data, that is, $\mathbf{x}_k$ *for some* k, and one of the signed samples, that is, $s_k\mathbf{x}_k$ *for some* k, respectively. When the Euclidean distance $\mathbf{x}_i - s_i\mathbf{c}_2^2$ goes to zero, the distance function $d(\mathbf{x}_i, s_i\mathbf{c})$ becomes $\log(\epsilon)$, which is an extremely small number. Based on this observation, we can find the global optimal solution to each of the problems. By enumerating each of the samples, the statistical mode problem (Equation (3)) can be expressed as

$$\min_{\mathbf{j}} \sum_{i=1}^{m} \log\left(\epsilon + \mathbf{x}_i^T\mathbf{x}_i - 2\mathbf{x}_i^T\mathbf{x}_j + \mathbf{x}_j^T\mathbf{x}_j\right). \tag{5}$$

For the modified l_1-PCA problem (Equation (4)), we must first remove the samples that were previously selected as optimal solutions to avoid selecting the same sample more than once. Equation (4) can then be expressed as

$$\begin{aligned}&\min_{\mathbf{j},s_i\in\{-1,1\}} \sum_{i=1} \log\left(\epsilon + \mathbf{x}_i^T\mathbf{x}_i - 2s_i\mathbf{x}_i^T\mathbf{x}_j + \mathbf{x}_j^T\mathbf{x}_j\right)\\ &= \min_{\mathbf{j}} \sum_{i=1} \log\left(\epsilon + \mathbf{x}_i^T\mathbf{x}_i - 2\left|\mathbf{x}_i^T\mathbf{x}_j\right| + \mathbf{x}_j^T\mathbf{x}_j\right).\end{aligned} \tag{6}$$

The equality holds because s_i is a binary variable and the minimum is attained only if $-2s_i\mathbf{x}_i^T\mathbf{x}_j = -2|\mathbf{x}_i^T\mathbf{x}_j|$. For the preceding two problems Eqs. (5) and ((6)), the dot products $\mathbf{x}_i^T\mathbf{x}_i$ and $\mathbf{x}_i^T\mathbf{x}_j$ can be pre-computed. The computation complexity for all of these m terms is thus $O(mD^2)$ for $\mathbf{x}_i \in R^D$. After that, we must only enumerate different combinations by performing addition and taking the logarithm. The computation complexity of finding the global optimal solution is very low, making it possible to obtain a fast and accurate solution. The full algorithm is described in Table 2.

Fig. 3 shows 10 video frames of the video sequence "Fall." The first four frames (i.e., Fig. 3(a)–(d)) are the wavering tree branches and comprise the background, while the last six frames (i.e., Fig. 3(e)–(j)) contain the moving truck and outliers and comprise the foreground. The number of frames that contain outliers is higher than the number of frames without outliers. After applying the algorithm in Table 2, video frame 2 is selected as the first component while video frame 3 is selected as the second component. Obviously, these frames comprise the video background. In this example, we can see that the proposed PMCA is robust to outliers even though the number of outliers exceeds 50% of the total samples. The PMCA is successful due to the characteristics of the statistical mode of the method. The four inliers are visually similar to one another, while the six outliers are relatively different from one another. This makes the four inliers form a compact group and become the "item" that appears most. However, the six outliers do not have this property and are dispersed. In this case, one of the four inliers is easily identified as a mode of the data. Figs. 4 and 5 show the first and second projected component of the Frames 552 to 556, Frames 1 to 4 and Frame 551 respectively. The first component represents the mostly appeared static background. We can see that the truck is removed from Frames 552 to 556 and only the trees and the road are remained. The second component represents the motion background that appears most. Some images in Fig. 5 show some white dots and they are the wavering tree branches and leaves. It is noted that Frame 2 is blank because it is the static background and has been removed in the first component. Moreover, the two components are required to be orthogonal to each other. That ensures the information extracted are different. So, these two components can represent a more complete dynamic background information.

To guarantee the projection vectors that are orthogonal to one another, we apply the orthogonalization procedure introduced by Kwak [47]. After obtaining the $(p-1)^{\text{th}}$ projection vector, the projected subsample is first projected to the following subspace:

$$\begin{aligned}\mathbf{x}_i(p) &= \left(\mathbf{I} - \mathbf{v}_{p-1}\mathbf{v}_{p-1}^T\right)\mathbf{x}_i(p-1) = \mathbf{x}_i(p-1)\\ &\quad - \left(\mathbf{v}_{p-1}^T\mathbf{x}_i(p-1)\right)\mathbf{v}_{p-1},\end{aligned} \tag{7}$$

Table 2
Algorithm to solve the relaxed l_0 problem and the modified l_1-PCA problem.

Input: Data $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_m]^T$
Output: Vector $\mathbf{v}_p$
Step 1: Compute $\mathbf{Y}^2 = \mathbf{Y}\mathbf{Y}^T$.
Step 2: If it is to find the first component $\mathbf{v}_1$, go to Step 3. Otherwise, go to Step 4.
Step 3: For j = 1 to m, compute $J_j = \sum_{i=1}^{m} \log(\epsilon + Y_{ii}^2 + Y_{jj}^2 - 2Y_{ij}^2)$. Go to Step 6.
Step 4: Remove the samples that are selected as optimal solutions for the previous projection vectors.
Step 5: For each j, compute $J_j = \sum_{i=1} \log(\epsilon + Y_{ii}^2 + Y_{jj}^2 - 2|Y_{ij}^2|)$.
Step 6: $j^* = \text{argmin}_j J_j$. The output is obtained by $\mathbf{v}_p = \mathbf{y}_{j^*}/\mathbf{y}_{j^*2}$.

Fig. 3. Illustration of the robustness of the proposed method to outliers. A truck passes in front of a tree shaken by the wind.

Fig. 4. The first projected component of the video frames shown in Fig. 3. Only static background is retained. The truck from Frames 552 to 556 are removed.

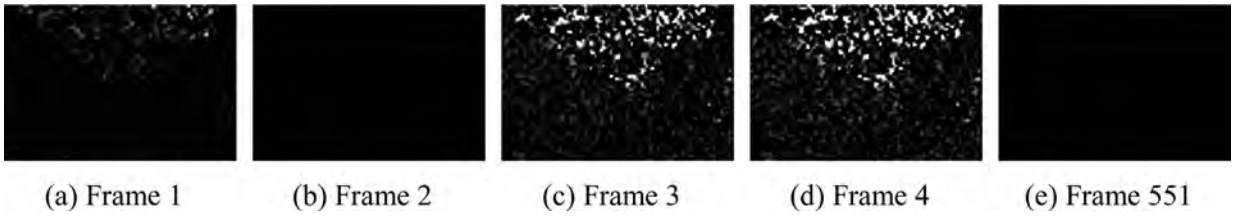

Fig. 5. The second projected component of the video frames shown in Fig. 3. Some of the tree shaken are extracted.

where $\mathbf{I}$ is the identity matrix, $\mathbf{v}_{p-1}$ is the $(p-1)^{th}$ principal component, and $\mathbf{x}_i(0) = \mathbf{x}_i$. We then treat the projected dataset $\mathbf{X}(p)$ as input data and apply the algorithm in Table 2 again to find the p^{th} projection vector $\mathbf{v}_p$. As $\mathbf{v}_p$ is a normalized sample drawn from the projected subsample $\{\mathbf{x}_i(p-1)\}_{i=1}^{m}$, we must have

$$\begin{aligned}\mathbf{v}_p^T\mathbf{v}_{p-1} &= \frac{\mathbf{x}_i(p)^T\mathbf{v}_{p-1}}{\mathbf{x}_i(p)_2} \\ &= \frac{\left(\mathbf{x}_i(p-1)^T\mathbf{v}_{p-1} - \left(\mathbf{v}_{p-1}^T\mathbf{x}_i(p-1)\right)\mathbf{v}_{p-1}^T\mathbf{v}_{p-1}\right)}{\mathbf{x}_i(p)_2} = 0,\end{aligned} \tag{8}$$

which means that $\mathbf{v}_p$ is orthogonal to $\mathbf{v}_{p-1}$. The same technique can be applied to show that $\mathbf{v}_p^T\mathbf{v}_j = 0$, *for* $j = 1, 2, \ldots p-2$. Thus, the obtained projection vectors are orthonormal to one another.

3.2.3. Identifying the foreground objects

To identify the foreground objects from a video sequence, we find the orthogonal complement of the subspace obtained by the projection vectors. The projection vectors $\mathbf{V} = [\mathbf{v}_1, \ldots \mathbf{v}_d]$ form a subspace that represents the most repetitive patterns of the video sequence. Its complement is the non-repetitive patterns, which are the outliers and thus the foreground objects. Given the k^{th} subsample obtained from the bootstrapping method, the absolute residual matrix $\mathbf{R}(k)$ can be computed by applying the orthogonal complement to video sequence $\mathbf{X}$. That is,

$$R_{it}(k) = \left|\left(\left(\mathbf{I} - \mathbf{V}\mathbf{V}^T\right)\mathbf{X}\right)_{it}\right|, \tag{9}$$

where $(\mathbf{X})_{it}$ is the $(i, t)^{th}$ entry of matrix $\mathbf{X}$. Fig. 6 shows an example of residual matrix $R(k)$ at the $(i, t)^{th}$ entry of matrix $\mathbf{X}$ with 45 bootstrap subsamples. To identify whether a pixel of a video frame is an outlier, we construct an outlier map O_{it} by considering the 10th percentile of the $(i, t)^{th}$ entry of all of the residual matrices $\mathbf{R}(k)$, for $k = 1, \ldots n_{boot}$.

$$O_{it} = \textit{the } 10\textit{th percentile of } R_{it}(k), \tag{10}$$

If O_{it} is too small, the i^{th} entry of the t^{th} video frame is similar to some of the most repetitive patterns. In other words, this entry has a high chance to be a video background.

3.3. The proposed algorithm

In this section, we combine all of the procedures introduced in the previous sections and explain the whole proposed method. The complete algorithm is shown in Table 3. We first rescale the video size to one of the three scales, that is, the original scale, 75% of the original size, or 50% of the original size, to incorporate multi-scale analysis into our proposed method. Multi-scale analysis has proved to be a robust tool for computer vision problems [48,49]. In each scale, we apply the bootstrapping method to generate n_{boot} subsamples. For each subsample, we find the projection vectors and obtain residual matrices $\mathbf{R}(k)$ for $k = 1, \ldots . n_{boot}$. As we need all of

Table 3
Proposed bootstrapping-based PMCA method.

Input: Video sequence **X**, number of projection vectors d, number of subsamples n_{boot}, subsample size m and size of the structural element S_{disk}
Output: Foreground map **F**
Step 1: Repeat the following steps with three different scales (s_{video}= 50%, 75%, 100%):
  Step 1a: Resize each of the video frames to be s_{video} of the original size and obtain $\mathbf{X}_s$.
  Step 1b: Set **T** as an empty three-dimensional array with size $m_l \times m_w \times m_d$.
  Step 1c: Repeat the following steps n_{boot} time
    Step 1c-(i) [Section 3.1]: Perform bootstrapping on video sequence $\mathbf{X}_s$ and obtain the k^{th} subsample. The size of each subsample is m. The details are shown in Table 1.
    Step 1c-(ii) [Sections 3.2.1 & 3.2.2]: To find d projection vectors, perform the following steps d times
      Step 1c-(ii)a: Find a projection vector by Table 2.
      Step 1c-(ii)b: Perform orthogonalization by Eq. (7).
    Step 1c-(iii) [Section 3.2.3]: Compute the residual matrix **R**(k) using Eq. (9) and take **T**(:, :, k) = **R**(k).
    Step 1c-(iv) [Section 3.2.3]: Sort array **T** from smallest to largest with respect to the third dimension of this matrix.
    Step 1c-(v) [Section 3.2.3]: Keep the smallest $(0.1 \times n_{boot})^{th}$ entries of sorted **T** and remove the rest of them.
  Step 1e [Section 3.2.3]: Find the outlier map **O** by taking the maximum of the sorted **T** in the third dimension.
  Step 1f [Section 3.2.3]: Resize the outlier map **O** by $1/s_{video}$ so that it has the same size as the original video.
Step 2 [Section 3.2.3]: Find outlier map $\mathbf{O}_{min}$ by $(\mathbf{O}_{min})_{ij}$ = minimum of the (i, j)th entry of the three outlier maps.
Step 3: Apply the morphological operation to outlier map $\mathbf{O}_{min}$ with a disk structural element and size S_{disk}. Output foreground map **F**.

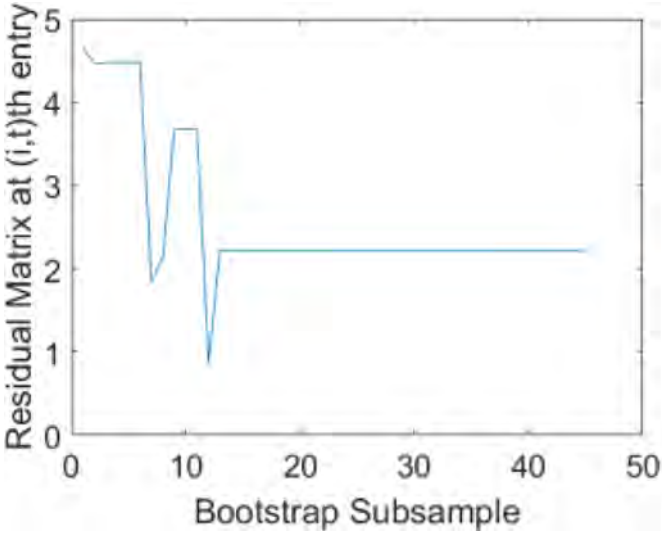

Fig. 6. An example of the residual matrix R(k) at the (i, t)th entry. [A larger version can be found in Appendix B.].

the entries within the 10th percentile of the residual matrices, we need only to record the $0.1 \times n_{boot}$ smallest values for each pixel of the video frame. This is carried out in Steps 1c-(iii) and 1c-(v). This can save much computer space. After that, we obtain the outlier map by taking the maximum of array **T** in the third dimension. This is because array **T** only records all of the values within the 10th percentile of the residual matrices. The largest value is the 10th percentile of the residual matrices. In Step 2, we combine the three outlier maps by taking the minimums of the respective entries of these three maps. This combined outlier map is resized to be the same size of the original video frame. Finally, the outlier map is polished by a morphological operation with a disk structural element and size S_{disk}.

4. Experiments

In this section, we compare the performance of the proposed method with that of state-of-the-art methods in addressing the background subtraction problem. We adopt an F-measure to assess the performance of different methods. The F-measure has been widely used as an evaluation metric in various background subtraction works [12–14,29]. It is defined as follows:

$$\mathrm{F-measure} = 2\frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \tag{11}$$

Precision and recall are calculated as follows:

$$\mathrm{precision} = \frac{\#\text{correctly classified foreground pixels}}{\#\text{pixels classified as foreground}} \tag{12}$$

$$\mathrm{recall} = \frac{\#\text{correctly classified foreground pixels}}{\#\text{foreground pixels in ground truth}} \tag{13}$$

The F-measure balances precision and recall and gives a score that shows the degree of similarity between the detected foreground objects and the ground truth foreground area. A larger value means a higher similarity. We compare the performance of the proposed method with that of 10 different methods, including TV-RPCA [13], TV1-RPCA [12], OMGMF [30], RPCA [8], Tensor-RPCA [14], Lag-SPCP-QN [50], MoG-RPCA [19], GreGoDec [51], RegL1-ALM [52], and MBRMF [18]. We adopt the default parameter settings for each of these methods. Some of the methods adopt different parameter settings for different videos, and we follow all of these settings. For the proposed method, we use the same set of parameters for all of the experiments. We set the number of projection vectors $d = 2$, the number of subsamples $n_{boot} = 50$, the subsample size $m = 10$, and the size of the structural elements $s_{disk} = 5$.

For the dataset, we adopt all nine videos from the I2R dataset [6],[1] five videos under the category of dynamic background from the CD.net dataset [53],[2] and five videos under the category of shadow from the CD.net dataset [53].[2] These datasets include various real-world scenes ranging from simple scenes (e.g., Hall, Bootstrap, Shopping-Mall) to illumination changes (e.g., Lobby) and dynamic backgrounds (e.g., Escalator, Curtain, Water-Surface). We adopt all of the video frames in each of our analyses. We resize some of the videos if they are too large.

Results for the I2R dataset: The quantitative results of different methods for the nine videos from the I2R dataset are shown in Table 4. Each value is the averaged F-measure taken over all of the foreground-annotated frames in the corresponding video. The best result for each video is shadowed. The proposed method obviously performs best in all nine videos; it even outperforms the second best method over 12% for the fountain video. For the rest of the videos, the proposed method is better than the second best method (around 2-5.4%). This shows the superiority of the proposed method to that of state-of-the-art methods. Fig. 7 (a larger version can be found in Appendix B (in the online supplementary materials)) shows the foreground detection results of different methods for the fountain and lobby videos. The fountain video has

[1] http://perception.i2r.a-star.edu.sg/bkmodel/bk index.html,
[2] http://changedetection.net.

Table 4
F-measure (%) comparison of different methods for all video sequences from the I2R dataset. [The recall measure can be found in Appendix C.].

	Campus	Fountain	Hall	Lobby	Curtain	Bootstrap	Shopping-Mall	Escalator	Water-Surface
TV-RPCA	0.8027	0.7465	0.6184	0.6723	0.7565	0.6776	0.6595	0.7354	0.8511
TV1-RPCA	0.7005	0.7309	0.5932	0.7148	0.785	0.6645	0.7393	0.7446	0.8137
OMGMF	0.4557	0.6225	0.575	0.7532	0.8623	0.6371	0.7173	0.5855	0.8737
RPCA	0.5564	0.7108	0.5568	0.7319	0.722	0.6741	0.7505	0.6987	0.424
Tensor-RPCA	0.5629	0.7107	0.6336	0.2054	0.8852	0.5147	0.6015	0.6294	0.8902
Lag-SPCP-QN	0.4438	0.6378	0.6695	0.6586	0.849	0.6329	0.7169	0.5992	0.8718
MoG-RPCA	0.4412	0.715	0.6164	0.758	0.6668	0.6181	0.7291	0.5461	0.7498
GreGoDec	0.4227	0.6491	0.6462	0.4383	0.8527	0.6313	0.7113	0.5987	0.8746
RegL1-ALM	0.4439	0.6363	0.6688	0.6618	0.8586	0.638	0.717	0.0894	0.8716
MBRMF	0.4534	0.6476	0.6391	0.7692	0.8362	0.6583	0.7154	0.6181	0.8733
Proposed	0.821	0.8675	0.7009	0.8234	0.9184	0.7072	0.792	0.7638	0.9382

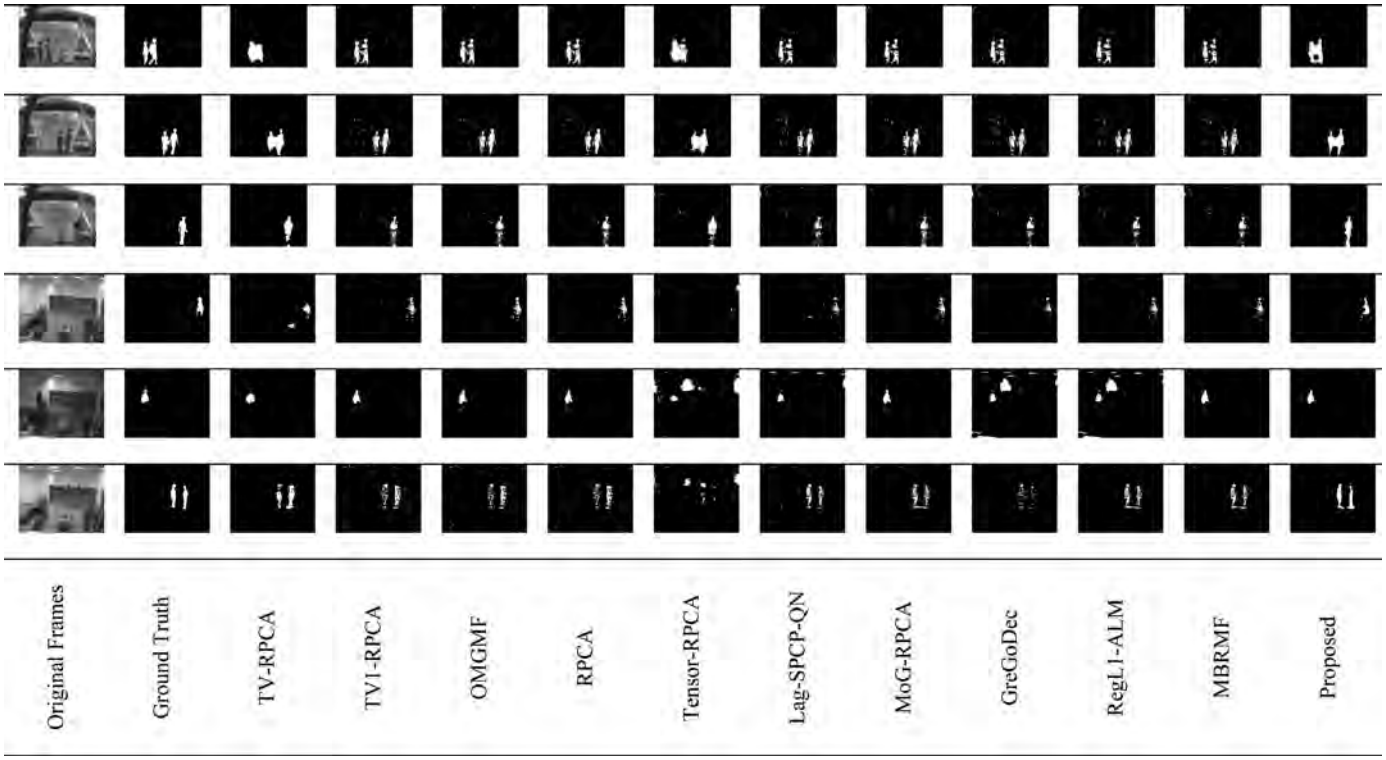

Fig. 7. Visual comparison of different methods for the Fountain Video (second to fourth row) and Lobby Video (last three tows).

a dynamic background, which is a waterfall fountain. For the fountain video results, most of the TV regularization-based methods (TV-RPCA, TV1-RPCA, and Tensor-RPCA) overly preserve the continuity of the foreground objects. Some of the objects are merged together and lose detailed information. To tune the parameter setting of these methods and obtain a better result, the whole optimization must be applied again. This can take much time. In contrast with the TV-based RPCA methods, the distributional approaches (such as MoG-RPCA and MBRMF) perform well only in the video sequences: Curtain, Shopping-Mall and Water-Surface. The reason may be that the assumed distributions of these models cannot fully reflect the key characteristics of the complex and dynamic background patterns. For MoG-RPCA, it adopts the mixture of Gaussian to model the complex and dynamic background patterns. Although mixture of Gaussian is proved to be a powerful tool, it is hard to precisely determine the number of Gaussian of the model. A non-optimal number of Gaussian may lead to a less accurate result. For the MBRMF, it adopts a Bayesian prior approach to model the parameters of the projected components and the complex and dynamic background patterns. It works well only if the Bayesian prior can model the situations well. However, for the complex and dynamic background patterns, the distributions can be in various forms. The assumed prior distribution may not fully reflect the real-world situations. For the proposed method, the results show that it can capture the spatial information of the foreground objects more accurately. For the lobby video, the video contains scenes of lights switching on and off. Some of the methods fail to detect the foreground objects and incorrectly detect the ceiling lamp as a foreground object. The proposed method and some state-of-the-art methods (such as OMGMF) can detect the foreground objects (fifth row) even though the moving person is relatively small in the video sequence. The performance of the proposed method is superior to that of state-of-the-art methods because bootstrapping the statistical mode of the video sequence captures almost all of the background video frames and constructs a more complete structure of the dynamic background. Although the fountain video has a water fountain in the background that confuses many of the methods, the proposed method captures all of the video frames that have different forms of the water fountain. When a video frame is similar to one of these background video frames, it is treated as background. Similarly, although the lobby video features lights changing, which has a great visual impact, the

Table 5
F-measure (%) comparison of different methods for all video sequences from the CD.net dataset. [The recall measure can be found in Appendix C.].

	Category: Dynamic Background					Category: Shadow				
	Canoe	Fall	Fountain01	Fountain02	Overpass	Back-Door	Bungalows	Bus-Station	Copy-Machine	People-in-Shade
TV-RPCA	0.7465	0.6073	0.2264	0.7311	0.5623	0.8276	0.5229	0.7276	0.6084	0.6541
TV1-RPCA	0.7309	0.6505	0.303	0.766	0.5946	0.8505	0.5249	0.6966	0.5953	0.6493
OMGMF	0.6225	0.3731	0.1705	0.6836	0.4725	0.7712	0.5911	0.711	0.6169	0.8063
RPCA	0.7108	0.544	0.2083	0.7292	0.5011	0.8377	0.5353	0.6895	0.5336	0.6654
Tensor-RPCA	0.7107	0.3713	0.2228	0.7652	0.5041	0.73	0.5101	0.6372	0.7562	0.7772
Lag-SPCP-QN	0.6378	0.3461	0.1504	0.6972	0.46	0.7294	0.599	0.7428	0.7377	0.8113
MoG-RPCA	0.715	0.3612	0.1554	0.7038	0.4592	0.7539	0.576	0.7226	0.8182	0.7086
GreGoDec	0.6491	0.3246	0.1531	0.725	0.4679	0.7327	0.5749	0.6926	0.7798	0.817
RegL1-ALM	0.6363	0.3457	0.1505	0.6973	0.4618	0.7313	0.5876	0.7462	0.7383	0.8167
MBRMF	0.6476	0.4174	0.1605	0.6939	0.4698	0.7683	0.5576	0.7005	0.7649	0.8032
Proposed	0.8675	0.713	0.3936	0.8841	0.5819	0.8791	0.5939	0.813	0.7369	0.8529

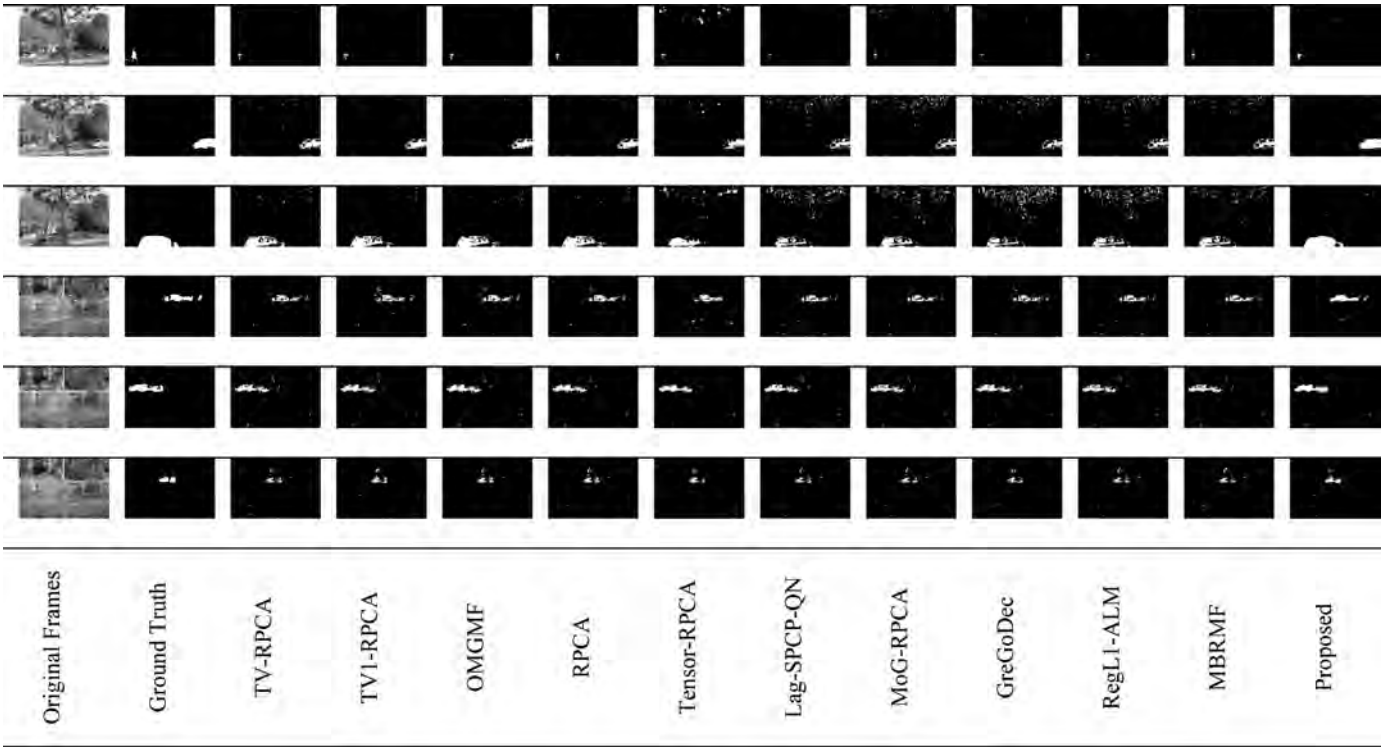

Fig. 8. Visual comparison of different methods for the Fall Video (second to fourth rows) and Fountain02 Video (last three rows).

proposed bootstrapping PMCA approach successfully captures both forms of the scene, that is, the video frames in which the lights are switched on and off. The video frames that are similar to any of these background frames are treated as background.

Results for CD.net dataset: The quantitative results for the different methods applied to the 10 videos from the CD.net dataset are shown in Table 5. The proposed method performs the best for seven different videos and second best for two videos. The proposed method performs more than 10% better than the second best method for the Canoe and Fountain02 videos and 2.8-9% better than the second best method for five different videos. Fig. 8 (a larger version can be found in Appendix B (in the online supplementary materials)) shows the detection results for the Fall and Fountain02 videos. Similar to the case of the I2R dataset, the proposed method outperforms the other methods. Its results look like the ground truth labels. For the Fall video, many of the methods incorrectly classify the wavering tree branches (third row of Fig. 8) as foreground objects while the proposed method successfully identifies them as part of the background. The success of the proposed method again owes to its ability to capture the complete patterns of the dynamic background, which is challenging for many methods. For the Fall video, the wavering tree branches have a cyclic pattern. The proposed method effectively identifies the video frames that have different forms of wavering branches. However, the proposed method does not perform as well as state-of-the-art methods for the Copy-Machine video. The misclassified pixels mainly belong to a person who spends a long time waiting for the printout. The person is nearly static for over 2,000 of the 3,400 frames of the video. As the proposed method adopts a subsampling scheme, it easily treats the waiting person as background.

Speed comparison: Fig. 9 shows the average computation time for different methods applied to 19 videos. The computation times for each video can also be found in Appendix D (in the online supplementary materials). GreGoDec and the proposed method have the lowest computation cost, and their average computation times are 3.77 s and 83.07 s, respectively. These are much lower than other methods such as MoG-RPCA and Lag-SPCP-QN, whose average computation times are 189.07 s and 189.97 s, respectively. Methods that involve singular value decomposition in each iteration of the optimization procedure, such as Tensor-RPCA, have a much larger computation time. The proposed method takes a much shorter computation time than the other methods for the following

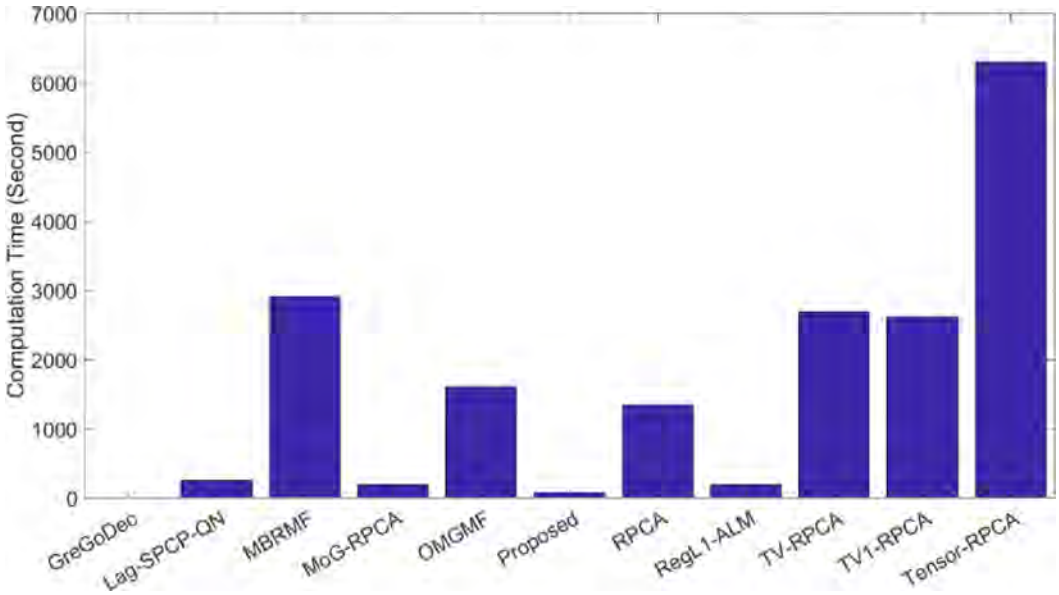

Fig. 9. Average computation time of different methods.

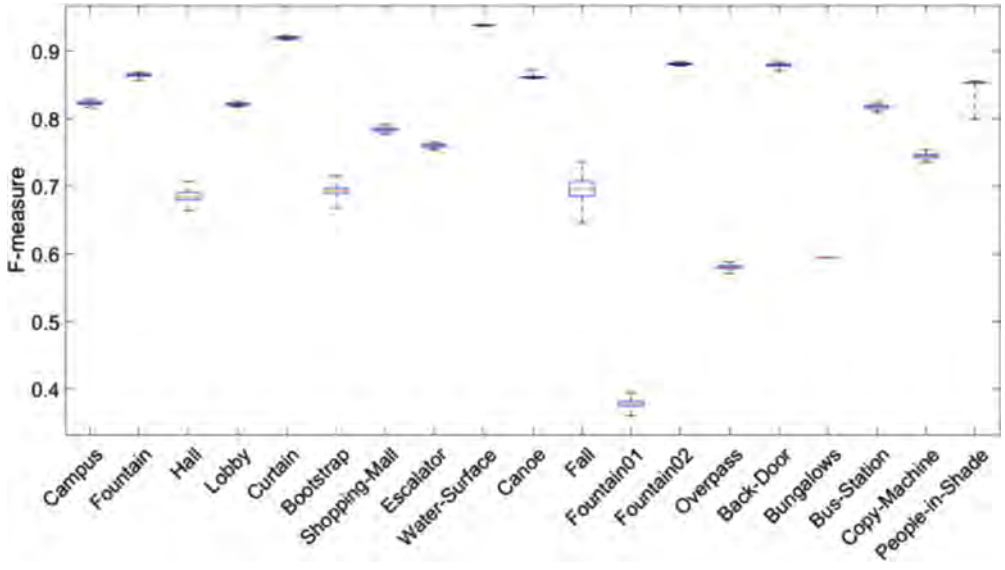

Fig. 10. Boxplot of the F-measure for the 19 videos under the 100 repeated runs. [A larger version can be found in Appendix B.].

reasons. The most computationally expensive step of the proposed method is performing PMCA for the subsamples. Section 3.2.2, explains that the fast exhaustive search must deal only with an $m \times m$ matrix after some simplifications. We set the size of a subsample to 10 to deal with a 10×10 matrix. This is not dependent on video size. The computation demand is much smaller than many current methods that compute the matrix multiplication many times, and the size of the matrix depends on the video size.

Empirical sensitivity analysis of the random effect of the proposed method: The proposed method adopts a bootstrapping method to generate a set of subsamples and produce the projection vectors. We now conduct a sensitivity analysis for the random effect of the proposed method. We apply the proposed method with exactly the parameter setting described at the beginning of this section to the 19 videos and repeat these experiments 100 times. In each run, we compute the F-measure. The boxplot of these 19 videos is shown in Fig. 10. The 5-point summary for each video can be found in Appendix E (in the online supplementary materials). For each box, it shows the minimum, first quantile, median, third quantile, and maximum of the F-measure for the 100 runs. A narrower vertical box means higher consistency across the 100 results. In most cases, the proposed method produces highly consistent results. Moreover, when we compare the worst performance of the proposed method (i.e., the minimum F-measure of the 100 runs for each video) with the second best methods as shown in Tables 4 and 5, we see that the proposed method is still the best in 12 different cases. When we consider the first quantile results, we see that the proposed method is the best in 16 different cases. These results show that the superiority of the proposed method is relatively insensitive to the random effect.

Results for vessel extraction: We also compare the performance of different PCA methods to the video sequence of X-ray coronary angiography [54,55]. The proposed method can detect the overall vessel structure more successfully. Because of the limited space, we refer the details to Appendix F.

5. Conclusion and future work

In this paper, we propose the use of a statistical bootstrapping method with statistical mode formulation to solve the background subtraction problem. Theoretically, the proposed bootstrapping method can capture almost any distribution without knowing the closed-form function. This differs from current methods that model the dynamic background with complicated noise using a closed-form probabilistic function. Moreover, we propose a new PCA method, PMCA, that can find the statistical modes of each of the subsamples. A statistical mode can capture the most repetitive pattern of the video sequence. By combining all of the statistical modes, the complicated and dynamic backgrounds can be captured. We test the proposed method using 19 different real-world video sequences from 2 popular datasets: I2R and CD.net. Our experiment results show that the proposed method performs the best in 16 cases and second best in 2 cases. We also propose a fast exhaustive search method to find the global optimal solution for the proposed PMCA. This fast method applies a simplification procedure that makes the optimization procedure independent of video size. This makes the proposed method more much computationally traceable than many other methods. Moreover, we perform sensitivity analysis for the random effect of the proposed method. Experiments show that the worst performance of the proposed method still performs better than state-of-the-art methods in 12 different cases, while the first quantile results of the proposed method are the best in 16 different cases.

A possible future work of this research is to solve the online background modelling problems. Current methodologies mainly rely on the incremental update that computes one frame at a time. Again, these methods usually assume the video sequence follows specific distributions. The bootstrapping technique introduced in this research work can model any distribution including those distributions without any closed-form expression. The proposed method should be a much more efficient method than the current incremental approach.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work described in this paper was partially supported by the grant from Research Grants Council of the Hong Kong Special Administrative Region, China (Projects UGC/FDS14/E03/14 and C1007-15 G). We would like to thank the Big Data and Artificial Intelligence Group of The Hang Seng University of Hong Kong for its support. We are thankful to the anonymous reviewers for their valuable comments that greatly helped to improve the paper.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2019.107153.

References

[1] X. Zhou, C. Yang, W. Yu, Moving object detection by detecting contiguous outliers in the low-rank representation, IEEE TPAMI 35 (2011) 597–610.
[2] C. Beleznai, B. Fruhstuck, H. Bischof, Multiple object tracking using local PCA, IEEE ICPR 3 (Aug. 2006) 79–82.
[3] W. Ge, Z. Guo, Y. Dong, Y. Chen, Dynamic background estimation and complementary learning for pixel-wise foreground/background segmentation, Pattern Recognit. 59 (2016) 112–125.
[4] D. Berjón, C. Cuevas, F. Morán, N. García, Real-time nonparametric background subtraction with tracking-based foreground update, Pattern Recognit. 74 (2018) 156–170.
[5] M. Babaee, D.T. Dinh, G. Rigoll, A deep convolutional neural network for video sequence background subtraction, Pattern Recognit. 76 (2018) 635–649.
[6] L. Li, W. Huang, I. Gu, Q. Tian, Statistical modeling of complex backgrounds for foreground object detection, IEEE TIP 13 (11) (2004) 1459–1472.
[7] N.J. McFarlane, C.P. Schofield, Segmentation and tracking of piglets in images, Machine Vision Appl. 8 (3) (1995) 187–193.
[8] E.J. Candes, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM 58 (3) (2011) 1–37.
[9] D. Meng, F. Torre, Robust matrix factorization with unknown noise, IEEE ICCV (2013) 1337–1344.
[10] X. Cao, Q. Zhao, D. Meng, Y. Chen, Z. Xu, Low-rank matrix factorization under general mixture noise distributions, IEEE TIP (2016) 4677–4690.
[11] P. Chen, N. Wang, N.L. Zhang, D. Yeung, Bayesian adaptive matrix factorization with automatic model selection, IEEE CVPR (2015).
[12] J. Xue, Y. Zhao, W. Liao, J. Chan, Total variation and rank-1 constraint RPCA for background subtraction, IEEE Access 6 (2018) 49955–49966.
[13] X. Cao, L. Yang, X. Guo, Total variation regularized RPCA for irregularly moving object detection under dynamic background, IEEE Trans. Cybernet. 46 (2016) 1014–1027.
[14] W. Cao, Y. Wang, J. Sun, D. Meng, C. Yang, A. Cichocki, Z. Xu, Total variation regularized tensor RPCA for background subtraction from compressive measurements, IEEE TIP 25 (2016) 4075–4090.
[15] H. Lu, X. Zhang, J. Qi, N. Tong, X. Ruan, M.H. Yang, Co-bootstrapping saliency, IEEE TIP 26 (2017) 414–425.
[16] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, 1st ed., Chapman & Hall/CRC, 1993.
[17] N. Wang, T. Yao, J. Wang, D. Yeung, A probabilistic approach to robust matrix factorization, Comp. Vis. (2012) 126–139.
[18] N. Wang, D. Yeung, Bayesian robust matrix factorization for image and video processing, IEEE ICCV (2013).
[19] Q. Zhao, D. Meng, Z. Xu, W. Zuo, L. Zhang, Robust principal component analysis with complex noise, ICML (2014).
[20] B. Lakshminarayanan, G. Bouchard, C. Archambeau, Robust Bayesian matrix factorization, AISTATS (2011).
[21] S.D. Babacan, M. Luessi, R. Molina, A.K. Katsaggelos, Sparse Bayesian methods for low-rank matrix estimation, IEEE TSP 60 (2012) 3964–3977.
[22] G. Zhang, Z. Yuan, Q. Tong, M. Zheng, J. Zhao, A novel framework for background subtraction and foreground detection, Pattern Recognit. 84 (2018) 28–38.
[23] D. Berjón, C. Cuevas, F. Morán, N. García, Real-time nonparametric background subtraction with tracking-based foreground update, Pattern Recognit. 74 (2018) 156–170.
[24] Y. Wang, C. Xu, C. Xu, D. Tao, Beyond RPCA: flattening complex noise in the frequency domain, AAAI Conf. Artif. Intell. (2017) 2761–2767.
[25] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999).
[26] X. Chen, Y. Cai, Q. Liu, L. Chen, Nonconvex lp-norm regularized sparse self-representation for traffic sensor data recovery, IEEE Access (2018).
[27] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Phys. D. 60 (1992) 259–268.
[28] X. Guo, X. Wang, L. Yang, X. Cao, Y. Ma, Robust foreground detection using smoothness and arbitrariness constraints, ECCV (2014) 535–550.
[29] H. Woo, H. Park, Robust asymmetric nonnegative matrix factorization, Computational and Applied Mathematics Reports, University of California, USA, 2014.
[30] C. Ding, H.X. He, H.D. Simon, On the equivalence of nonnegative matrix factorization and spectral clustering, Proc. SIAM Int. Conf. Data Mining (2005) 606–610.
[31] T. Yu, L. Wang, C. Guo, H. Gu, S. Xiang, C. Pan, Pseudo low rank video representation, Pattern Recognit. 85 (2019) 50–59.
[32] L. Qi, X. Lu, X. Li, Exploiting spatial relation for fine-grained image classification, Pattern Recognit. 91 (2019) 47–55.
[33] X. Lu, Y. Chen, X. Li, Hierarchical recurrent neural hashing for image retrieval with hierarchical convolutional features, IEEE TIP Vol.27 (2018) 106–120.
[34] X. Lu, W. Zhang, X. Li, A hybrid sparsity and distance-based discrimination detector for hyperspectral images, IEEE Trans. Geosci. Remote Sens. 56 (2018) 1704–1717.
[35] H. Yong, D. Meng, W. Zuo, L. Zhang, Robust online matrix factorization for dynamic background subtraction, IEEE TPAMI 40 (2018) 1726–1740.
[36] J. He, L. Balzano, A. Szlam, Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video, CVPR (2012) 1568–1575.
[37] J. Xu, V.K. Ithapu, L. Mukherjee, J.M. Rehg, V. Singh, GOSUS: Grassmannian online subspace updates with structured-sparsity, ICCV (Dec. 2013) 3376–3383.
[38] H. Guo, C. Qiu, N. Vaswani, An online algorithm for separating sparse and low-dimensional signal sequences from their sum, IEEE TSP 62 (Aug. 2014) 4284–4297.
[39] P. Narayanamurthy, N. Vaswani, Medrop: memory efficient dynamic robust PCA, IEEE ICASSP (2017).
[40] P. Rodriguez, B. Wohlberg, Incremental principal component pursuit for video background modeling, J. Math. Imag. Vis. 55 (2016) 118.

[41] P. Rodríguez, B. Wohlberg, A MATLAB implementation of a fast incremental principal component pursuit algorithm for video background modeling, IEEE ICPR (2014) 3414–3416.
[42] M. Villamizar, J. Andrade-Cetto, A. Sanfeliu, F. Moreno-Noguer, Bootstrapping Boosted Random Ferns for discriminative and efficient object classification, Pattern Recognit. 45 (2012) 3141–3153.
[43] V.V. Saradhi, M.N. Murty, Bootstrapping for efficient handwritten digit recognition, Pattern Recognit. 34 (2001) 1047–1056 ppl.
[44] J.N. Myhre, K. Ø. Mikalsen, L. S., R. Jenssen, Robust clustering using a kNN mode seeking ensemble, Pattern Recognit. 76 (2018) 491–505.
[45] Z. Zhang, Y. Xu, J. Yang, X. Li, D. Zhang, A survey of sparse representation: algorithms and applications, IEEE Access (2015).
[46] W. Rudin, Real and Complex Analysis, 3rd ed., McGraw-Hill Education, 1986.
[47] N. Kwak, Principal component analysis based on l-1-norm maximization, IEEE TPAMI 30 (2008) 1672–1680.
[48] M. Li, Fast translation invariant multiscale image denoising, IEEE TIP 24 (2015) 4876–4887.
[49] R. Soulard, P. Carré, Characterization of color images with multiscale monogenic maxima, IEEE TPAMI 40 (2017) 2289–2302.
[50] A. Aravkin, S. Becker, V. Cevher, P. Olsen, A variational approach to stable principal component pursuit, in: Conference on Uncertainty in Artificial Intelligence, 2014, pp. 32–41.
[51] T. Zhou, D. Tao, Greedy bilateral sketch, completion and smoothing, International Conference on Artificial Intelligence and Statistics, 2013.
[52] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, M. Okutomi, Practical low-rank matrix approximation under robust L1-norm, CVPR (2012) 1410–1417.
[53] N. Goyette, P.M. Jodoin, F. Porikli, J. Konrad, P. Ishwar, Change detection.net: a new change detection benchmark dataset, CVPR (2012) 1–8.
[54] M. Jin, R. Li, J. Jiang, B. Qin, Extracting contrast-filled vessels in X-ray angiography by graduated RPCA with motion coherency constraint, Pattern Recognit. 63 (2017) 653–666.
[55] B. Qin, M. Jin, D. Hao, Y. Lv, Q. Liu, Y. Zhud, S. Ding, J. Zhao, B. Fei, Accurate vessel extraction via tensor completion of background layer in X-ray coronary angiograms, Pattern Recognit. 87 (2019) 38–54.

Benson, S. Y. LAM. Dr. Lam is currently an Assistant Professor in the Department of Mathematics and Statistics at the Hang Seng University of Hong Kong. He received both BSc and MPhil in Mathematical Science from the Department of Mathematics at Hong Kong Baptist University. Upon his completion of Ph.D. degree at City University of Hong Kong, he joined Griffith University (Brisbane, Australia) as a postdoctoral researcher for two years.

Amanda, M. Y. Chu. Dr. Chu is currently an Assistant Professor in the Department of Social Sciences, The Education University of Hong Kong. She received her Ph.D. from The University of Hong Kong, MBA from The Chinese University of Hong Kong, and BSocSc in Statistics from The University of Hong Kong. Her current research interests include data privacy, information security, risk management, and applied statistics. Prior to studying for her Ph.D., Dr. Chu was an industry consultant for over 8 years.

Hong Yan. Hong Yan received his Ph.D. degree from Yale University. He was professor of imaging science at the University of Sydney and currently is Chair Professor of Computer Engineering and Wong Chung Hong Professor of Data Engineering at City University of Hong Kong. Professor Yan was elected an IAPR fellow for contributions to document image analysis and an IEEE fellow for contributions to image recognition techniques and applications. He received the 2016 Norbert Wiener Award from IEEE SMC Society for contributions to image and biomolecular pattern recognition techniques.