Block-Sparse RPCA for Salient Motion Detection

Zhi Gao, Loong-Fah Cheong, and Yu-Xiang Wang

Abstract—Recent evaluation [2], [13] of representative background subtraction techniques demonstrated that there are still considerable challenges facing these methods. Challenges in realistic environment include illumination change causing complex intensity variation, background motions (trees, waves, etc.) whose magnitude can be greater than those of the foreground, poor image quality under low light, camouflage, etc. Existing methods often handle only part of these challenges; we address all these challenges in a unified framework which makes little specific assumption of the background. We regard the observed image sequence as being made up of the sum of a low-rank background matrix and a sparse outlier matrix and solve the decomposition using the Robust Principal Component Analysis method. Our contribution lies in dynamically estimating the support of the foreground regions via a motion saliency estimation step, so as to impose spatial coherence on these regions. Unlike smoothness constraint such as MRF, our method is able to obtain crisply defined foreground regions, and in general, handles large dynamic background motion much better. Furthermore, we also introduce an image alignment step to handle camera jitter. Extensive experiments on benchmark and additional challenging data sets demonstrate that our method works effectively on a wide range of complex scenarios, resulting in best performance that significantly outperforms many state-of-the-art approaches.

Index Terms—Block-sparse RPCA, salient motion, dynamic background, camera jitter

1 Introduction

Our society has invested massively in the collection and processing of data of all kinds, on scales unimaginable until recently. The web also has an enormous collection of live cameras that capture images of roads, beaches, cities, buildings, and forests. Images from these cameras are a vast untapped resource of information about the world and the way it changes over time. For example, they might be used for surveying animal populations, monitoring coastal erosion, and security. Change detection will be a key component of these data processing activities. Despite much effort in this direction, a recent evaluation of major techniques for video surveillance [2] showed that hardly any approach can reach more than 50 percent precision at recall level higher than 90 percent. Designing an algorithm that is robust under a wide variety of scenes encountered in complex real-life applications remains an open problem.

For cameras that are mounted and are more or less stationary, background subtraction is a major class of technique used to detect changes or moving objects. Essentially, in such methods, video frames are compared with a background model; changes are then identified as the foreground. Various methods have been used to model the background, ranging from simple thresholding [32] to various forms of parametric approach such as a single Gaussian [6], [16] or a mixture of Gaussians [28], and extensions of such Gaussian mixture model [10], [14], [17]. Without attempting to be exhaustive, other methods include kernel density estimate (KDE) using either Gaussian kernels [9], [27], variable-bandwidth kernels [22] or step kernels [1], [38], histogram [36], neural networks [20], Markov random fields (MRF) models [26], block correlation [21], and codebook model [18]. For cameras that are moving or experiencing significant jitter, a preprocessing step of image alignment is needed. For a more detailed discussion of some of these techniques, readers can refer to a recent survey [3].

The reason why most of these methods fail to work well in realistic complex situation is that these methods often make overly restrictive assumptions about the background. In reality, the background itself can have complex changes. It might contain motion such as those caused by ripples on a lake, or swaying vegetation, which can cause false alarms. The motion of these backgrounds can be larger than that of the foreground. There could be sudden illumination change caused by cloud cover, causing complex intensity and shadow variation, or more gradual illumination change caused by the movement of the sun. During dawn and dusk hours, the image quality can be poor due to the low light condition. In view of these complex factors, it is very difficult to model the background well. Training-based methods also assume the availability of training clips with no foreground motions.

In this paper, we handle all these challenges by making very little specific assumptions about the background. The only assumption made about the background is that any variation in its appearance (whether caused by intensity change, or non-stationary background) is highly constrained and can be captured by the low rank condition of a suitably formulated matrix. In its simplest form, we say that a $m \times n$ matrix $\boldsymbol{M}$ composed of the observed vectorized image frames (i.e., $m = \text{number of rows} \times \text{number of columns}$, $n = \text{number of frames}$) can be decomposed into a low-rank matrix $\boldsymbol{L}$ representing the background, and a sparse matrix

- *Z. Gao is with the Interactive and Digital Media Institute, National University of Singapore, Singapore 119613. E-mail: gaozhinus@gmail.com.*
- *L.-F. Cheong and Y.-X. Wang are with the Electrical and Computer Engineering Department, National University of Singapore, Singapore. E-mail: {eleclf, wangyx}@nus.edu.sg.*

Manuscript received 6 Mar. 2013; revised 23 Feb. 2014; accepted 16 Mar. 2014. Date of publication 31 Mar. 2014; date of current version 10 Sept. 2014. Recommended for acceptance by C. Sminchisescu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2014.2314663

$\boldsymbol{S}$ consisting of the moving objects treated as sparse outliers. Detecting the moving objects amounts to solving the following problem:

$$\min_{\boldsymbol{L},\,\boldsymbol{S}} \operatorname{rank}(\boldsymbol{L}) + \lambda ||\boldsymbol{S}||_0, \quad \text{s.t. } \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}, \tag{1}$$

where λ is a regularizing parameter. This is also known as the Robust Principal Component Analysis (RPCA) problem [4]. Recently, there has been a spate of methods proposing the use of such low-rank constraint [11], [34], [35]. In the next section, we will first discuss the inadequacy of such formulation, and then introduce the necessary steps that will lead to substantially better results than other RPCA related formulations and other state-of-the-art background subtraction techniques, as has been shown in the conference version of our work [12]. In this paper, we also extend our work to handle more complex foreground motion and camera jitter, and we perform a thorough evaluation of the proposed method on both the SABS [2] and the CDnet [13] data sets.

2 Background and System Overview

In the earliest models using low rank matrix to represent background [4], [7], no prior knowledge on the spatial distribution of outliers was considered. In real videos, the foreground objects are usually spatially coherent clusters. Thus, contiguous regions should be preferably detected. Such prior has been incorporated through the MRF prior [37]; however the result of imposing such smoothness constraint (even with the so-called discontinuity preserving prior such as those based on Potts model) is that the foreground region tends to be over-smoothed. For instance, the detailed silhouette of the hands and legs of a moving person is usually sacrificed in favor of a more compact blob. Our idea is related to the so-called block-sparsity or joint-sparsity measures to incorporate spatial prior. However, these works [8], [29] typically assume that the block structure is known. Rosenblum et al.'s [25] method does not require prior knowledge on the block size and block location, which are instead detected by iteratively alternating between updating the block structure of the dictionary and updating the dictionary atoms to better fit the data. Nevertheless, both the number of blocks and the maximal block size are assumed to be known. In [15], [24], the sparsity structure is estimated automatically. However, in [15], parameter tuning is required to control the balance between the sparsity prior and the group clustering prior for different cases, and both these algorithms share the same limitation that training sequences composed of clean background are required. In contrast, our method estimates the sparsity structure automatically without a separate training phase.

Before describing how to automatically detect the blocks containing moving objects, we want to discuss the issue of scale. The scale issue is present in the preceding RPCA related formulation because there is no one value of λ, the regularizing parameter, that can handle foreground objects of all kind of sizes (λ controls the amount of outliers in the RPCA decomposition, and thus is related to the scale issue). This issue is in fact a perennial challenge in many segmentation problems. As an example, in the well-known Normalized Cut algorithm, there often cannot be a single correct segmentation of an image unless it has a single prominent object. Let us take an example of the scene shown in Fig. 5, in which the tree is much larger in size than the human, and its apparent image motion is also larger due to its proximity to the camera. For such a scene with prominent objects appearing at significantly different scales, having a single global parameter for segmenting the scene (whether in the sense of image segmentation, or in the present case, motion segmentation) is not even meaningful. While the block-sparsity approach to a certain extent can relieve this scale problem (by having blocks of different sizes), it does not fundamentally remove the problem, especially when there is a lot of large background motion.

The root of this problem lies in that the precise definition of the foreground target is intricately linked with the object of interest in the scene (i.e., one's purpose) and can be well defined only if the object of interest or its salient characteristics is known to us. However, knowing about the object of interest even before segmenting the scene seems to make the problem as one of many chicken-egg problems in computer vision, as we usually need to segment the scene to recognize the objects in it. So, how can we identify an object and its probable size even before segmenting it?

Clearly this must involve a feedback process, either implicitly or explicitly. In this paper, we put forth a hierarchical two-pass process to solve the aforementioned problems. The first-pass RPCA rapidly identifies the likely regions of foreground in a sub-sampled image. A simple motion consistency scheme is then used to measure the motion saliency of these foreground regions. Then in the second pass, a block-sparse RPCA imposes the spatial coherence of foreground objects in the outlier matrix $\boldsymbol{S}$, with the λ value set according to the motion saliency estimated in the first pass. Taking into account the motion saliency and the block-sparse structure of the outlier matrix $\boldsymbol{S}$ makes the foreground detection robust against the clutter caused by the background motion, and largely invariant to object size, allowing us to return crisply defined foreground regions. As opposed to formulating the whole problem into a single optimization function (as in the case of [15], [24]), we favor the explicit modeling of this feedback process. Not only this achieves greater modularity of different processes and ensures convergence, it also allows greater flexibility in the design of the motion saliency measure (other domain specific constraints can be readily accommodated too). This gives us a greater advantage when decomposing scenes with complex foreground and background motion, as can be seen from the experimental results later. Note that our strategy is also distinct from recent region segmentation method [5] which uses mid-level Gestalt properties such as convexity to rank multiple region hypothesis produced by different spatial scales. We use the estimated motion saliency to set a more fine-tuned λ value in each block in the second pass, rather than just using it to rank hypothesis. Another merit of having a two-pass process is that we can incorporate in an integral manner an image alignment step similar to RASL [23] in the first pass. Since the first pass is carried out over a sub-sampled image, the amount of computation is significantly reduced.

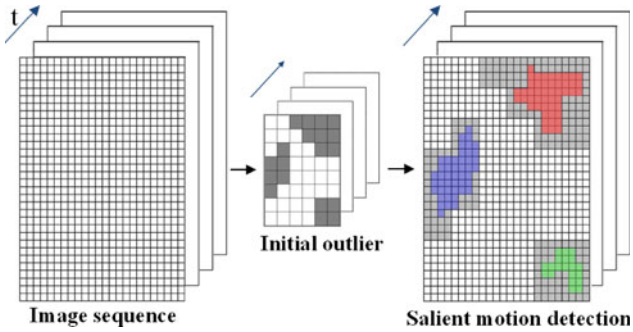


Fig. 1. Middle: Initial outlier detection via a first-pass RPCA on a sub-sampled sequence. Each grid in this figure corresponds to a 4×4 block in the full resolution image. Right: The remaining computation is carried out in original resolution; gray pixels represent background objects performing non-salient motion; those color pixels denote foreground objects with salient motions.

We test our algorithm on background with a wide range of dynamic texture motions with varying magnitude, and also on different sizes of foreground objects. Other challenging conditions as categorized in [2] and [13] (illumination change, high noise in dim environment, artifacts in thermal images, intermittent motions, etc.) are also tested with the sequences provided therein. In all cases other than those categories which we do not explicitly tackle (such as shadows), we are able to achieve accurate detection and clean delineation of targets. Our algorithm emerges as the overall winner, with consistent performance across all categories that significantly improves over those achieved by most state-of-the-art techniques, including DPGMM [14] which is just released at the time of writing and currently the best technique according to the evaluation criteria on CDnet.

3 Our Algorithm

3.1 First-Pass RPCA

As discussed in the preceding section, our proposed approach is based on a two-pass RPCA process. First we make a rapid weak identification in the form of rough region(s) of interest by performing a first-pass RPCA on a scaled-down low resolution sequence (sub-sampled spatially at a four to one ratio). This step is based on a simple convex relaxation of equation (1):

$$\min_{\boldsymbol{L},\boldsymbol{S}} \ ||\boldsymbol{L}||_* + \lambda||\boldsymbol{S}||_1 \quad \text{s.t.} \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}, \tag{2}$$

where $||\boldsymbol{L}||_*$ is the nuclear norm of matrix $\boldsymbol{L}$, the sum of its singular values, and λ set at a value that ensures no genuine foreground regions will be missed. The inexact augmented Lagrange multiplier (ALM) method [19] is used to solve this problem. We find that the recommended value of $\lambda = 1/\sqrt{max(m,n)}$ (where $m \times n$ are the dimensions of $\boldsymbol{M}$) is adequate to identify all foreground regions, including possibly many background regions.

As we find in our experiments later, while the low-rank formulation of the background matrix is generally effective in absorbing many natural variations in the background (such as illumination change), the full power of the RPCA framework to achieve accurate decomposition can only be realized if we have a more subtle mechanism to handle the aforementioned issue of scale and to capture the salient aspect of the expected foreground motions. As it is, not only significant number of non-stationary background points are being deposited in the outlier matrix, faint trace of the foreground motion is often also retained in the background matrix (see the ghostly presence of the walking person in the inset of Fig. 5). It is evident that there is no single λ that can achieve a clean separation of the foreground and background regions.

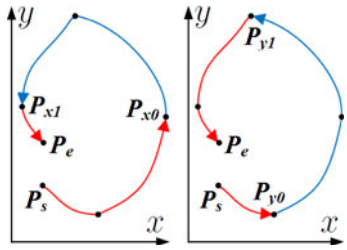


Fig. 2. Trajectory P_sP_e partitioned according to two criteria: when the curve changes direction in x (left) and in y (right). The points P_{xi} (left) and P_{yi} (right) represent the partition points.

Referring to the middle figure of Fig. 1, all those non-white pixels are the outliers estimated via the first-pass RPCA. These outliers typically comprise both background and foreground objects. Before we discuss how we identify those blocks with genuine outliers in Section 3.3, we have to first discuss the necessary alignment step if camera jitter is an issue.

3.2 First Pass with Image Alignment

Camera jitters caused by strong wind, passing vehicles and so on can severely impair the performance of our RPCA based algorithm if it is not properly compensated for. Following the work of RASL [23], we can incorporate a similar alignment process seamlessly into the first pass presented above.

Suppose we are given n images $I_1^0, \ldots I_n^0$ of some scene captured by a jittering camera. We can model the alignment process as a 2D parametric transform $\tau_1, \ldots, \tau_n \in \mathbb{R}^2$ acting on the two-dimensional domain of the images respectively, such that the i-th frame after alignment is denoted as $I_i^0 \circ \tau_i$. If the sequence is well-aligned, it should exhibit good low-rank structure, up to some sparse outliers. We therefore search for a set of transformations $\tau = \{\tau_1, \ldots, \tau_n\}$ such that after the sparse outliers are accounted for, the rank of the transformed sequence should be as small as possible. Thus, instead of Equation (2), we now have Equation (3) where, to facilitate description, we have used $\boldsymbol{D} \circ \tau$ as shorthand for $[vec(I_1^0 \circ \tau_1)|\ldots|vec(I_n^0 \circ \tau_n)] \in \mathbb{R}^{m \times n}$ and vec is an operator that stacks an image as a vector.

$$\min_{\boldsymbol{L},\boldsymbol{S},\tau} \ ||\boldsymbol{L}||_* + \lambda||\boldsymbol{S}||_1 \quad \text{s.t.} \quad \boldsymbol{D} \circ \tau = \boldsymbol{L} + \boldsymbol{S}. \tag{3}$$

The main difficulty in solving (3) is the nonlinearity of the constraint $\boldsymbol{D} \circ \tau = \boldsymbol{L} + \boldsymbol{S}$. Following [23], we approximate this constraint by linearizing around the current estimate of τ. That is, we have $\boldsymbol{D} \circ (\tau + \triangle\tau) \approx \boldsymbol{D} \circ \tau + \sum_{i=1}^{n} J_i \triangle \tau_i \epsilon_i^T$, where $J_i \doteq \frac{\partial}{\partial \zeta} vec(I_i^0 \circ \zeta)|_{\zeta=\tau_i}$ is the Jacobian of the i-th image with respect to the transformation

parameters τ_i and $\{\epsilon_i\}$ denotes the standard basis for $\mathbb{R}^n$. This leads to a convex optimization problem in the unknowns $\boldsymbol{L}$, $\boldsymbol{S}$, $\triangle\tau$:

$$\min_{\boldsymbol{L},\boldsymbol{S},\triangle\tau} ||\boldsymbol{L}||_* + \lambda||\boldsymbol{S}||_1 \quad \text{s.t. } \boldsymbol{D}\circ\tau + \sum_{i=1}^{n} J_i\triangle\tau_i\epsilon_i^T = \boldsymbol{L}+\boldsymbol{S}. \tag{4}$$

To find the (probably local) minimum, we repeatedly linearize about the current estimate of τ and solve a sequence of convex programs of the form of (4) until convergence.

This alignment step could be potentially costly due to its iterative linearizations. However, having the alignment in the first pass results in significant savings since we are only dealing with a sub-sampled image (60×80 in our experiments); down-sampling also facilitates the convergence of algorithm and avoids being trapped in a local minimum. In this work, we find that a simple translation model for the transformation τ is good enough to handle those camera jitters encountered in our experimental sequences. Our first pass RPCA operation is summarized as Algorithm 1.

Algorithm 1 First pass RPCA with alignment and outlier detection.

Input:
Image sequence $I_1^0,\ldots,I_n^0 \in \mathbb{R}^{w\times h}$, all transformations $\tau_1,\ldots,\tau_n$ initialized to $(0,0)^T$, default weight $\lambda > 0$;

Output:
Solution $\boldsymbol{L},\boldsymbol{S},\tau$ of equation (3), and a well-aligned image sequence $I_1,\ldots,I_n \in \mathbb{R}^{w\times h}$;

1: Downsample $I_1^0,\ldots,I_n^0$ to obtain $\hat{I}_1^0,\ldots,\hat{I}_n^0 \in \mathbb{R}^{\frac{w}{4}\times\frac{h}{4}}$;
2: **while** not converged **do**
3: compute Jacobian matrices w.r.t. transformation:
$J_i \leftarrow \frac{\partial}{\partial\zeta}\left(\frac{vec(\hat{I}_i^0\circ\zeta)}{||vec(\hat{I}_i^0\circ\zeta)||}\right)\Big|_{\zeta=\tau_i}, i=1,\ldots,n$
4: warp and normalize the images:
$\boldsymbol{D}\circ\tau \leftarrow [\frac{vec(\hat{I}_1^0\circ\tau_1)}{||vec(\hat{I}_1^0\circ\tau_1)||}|\ldots|\frac{vec(\hat{I}_n^0\circ\tau_n)}{||vec(\hat{I}_n^0\circ\tau_n)||}]$;
5: solve the linearized convex optimization problem (4):
$(\boldsymbol{L},\boldsymbol{S},\triangle\tau) \leftarrow \underset{\boldsymbol{L},\boldsymbol{S},\triangle\tau}{argmin}\ ||\boldsymbol{L}||_* + \lambda||\boldsymbol{S}||_1$
$subj \quad \boldsymbol{D}\circ\tau + \sum_{i=1}^{n} J_i\triangle\tau_i\epsilon_i^T = \boldsymbol{L}+\boldsymbol{S}$
6: update transformations:
$\tau \leftarrow \tau + \triangle\tau$
7: **end while**
8: Apply transformations τ on sequence $\hat{I}_1^0,\ldots,\hat{I}_n^0$, followed by upsampling to obtain $I_1,\ldots,I_n \in \mathbb{R}^{w\times h}$.

3.3 Motion Saliency Estimation

The likelihood of a block generated by the first pass containing genuine foreground motions is measured by the saliency of its motion, based on a method modified from [33]. The basic idea is to track all pixels within the blocks detected in the first pass RPCA via dense optical flow. Only those trajectories whose directions are consistent, or at least consistent for a certain minimum amount of duration, are deemed to be salient and retained as likely candidates of foreground motions. A simple block merging step is then carried out to group those spatially connected 4×4 salient blocks into a larger rectangular block that encompasses all of them (see Fig. 3), forming the block structure for the second-pass RPCA process.

Our motion saliency estimation method differs from that of Wixson [33] in the following ways. Instead of using the KLT tracker, we applied the "Classic + NL" algorithm [30] to obtain dense optical flow, from which we can assess the motion consistency of an entire region rather than just a set of sparse features. It also suits our purpose that this algorithm is robust to ambient illumination change. Another difference with [33] is that we attempt to track a point for as long as possible, and compute the motion consistency based on the entire trajectory, rather than just on a window of about ten frames. This is important for handling "periodic" background motion whose period might be longer than 10 frames (for the details of the tracking procedure, please refer to the Appendix). Lastly, and most importantly, our definition of what constitutes salient motion also differs from those of [33] as we want to capture a bigger class of foreground motions which are not necessarily always moving in one direction (more of this in the next paragraph). Note that the tracking is done only at the pixel level. There is no tracking at the block level; thus, the saliency of the block is entirely dependent on the saliency of constituent trajectories passing through the block at that instant.

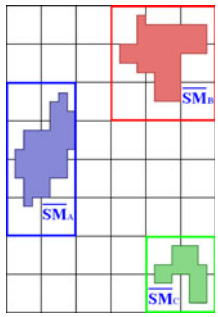


Fig. 3. Foreground regions with different motion saliency measure $\overline{SM}_A, \overline{SM}_B, \overline{SM}_C$. Here, each grid cell is not a pixel but a 4×4 block.

We now describe the criteria for motion saliency. Denoting the successive 2D image position of points along the lth trajectory from the j_lth frame to the k_lth frame ($j_l, k_l \in [1, n]$) as $X_{l,j_l}, X_{l,j_l+1}, \ldots X_{l,k_l}$, we first remove those trajectories that are too short, specifically, $k_l - j_l \leq 10$. We then check for the following two conditions. To fulfil Condition 1, we require that the time spent moving in one direction accounts for more than 80 percent of the entire trajectory (this is similar to Wixson's definition of motion consistency [33]).[1] If this condition is met, the saliency measure SM is defined to be proportional to the displacement traveled by that pixel. This definition of saliency measure serves to further enhance the detectability of those foreground motion moving in a slow but consistent manner for a sustained period of time. In general, slow motion causes a smaller rank change to the matrix L and is more liable to be missed if its associated λ is not suitably compensated. Conversely, inconsistent motions of the background that result in small local displacement will be further discounted if

1. See Algorithm 2 for details. Note that we just determine the motion consistency along the horizontal and vertical directions of movement for simplicity. The reason is that it could be quite difficult to determine the main direction of movement sometimes, such as in the case of evasive motions, where there might not be an overall direction of move. The threshold of 0.01 pixel in step 6 is empirically determined to remove those small spurious motion which could be generated by the propagation mechanism in dense optical flow estimation algorithm.

they are not already removed in the preceding pass. If the first condition is not met, we nevertheless have Condition 2 that would also qualify a motion as salient. Among the many foreground motions of interest, there might be many of them with a significant amount of back and forth movements (i.e., their motions are inconsistent); examples include sport games, activities in confined spaces like kitchen and playroom. What distinguishes them from those quasi-periodic, inconsistent motions found in the background (e.g., waves, foliage movements) is that the duration of one such "back" or "forth" episode is normally longer. Here we declare that the trajectory motion is salient if the aforementioned episode duration is longer than three seconds. The saliency measures are then normalized across the two conditions. Specifically, the saliency measure of a trajectory obtained from condition 2 is adjusted to be between the minimum and the maximum of the SM values returned from Condition 1, the exact amount determined by the relative episode duration (see step 12 in Algorithm 2 for details). Finally, for block i, we average the SM of all consistent trajectories with $SM_l > 0$ that pass through this block, and denote the average as $\overline{SM}_i$.

Algorithm 2 Motion Saliency Computation.

Input: All m trajectories $X_{l,j_l}, X_{l,j_l+1}, \ldots, X_{l,k_l}$, $l = 1, \ldots, m$;
Output: Saliency measure of the trajectories: SM_l, $l = 1, \ldots, m$;
1: **for** l=1; $l <= m$; l++
2: //Lines 3 initializes the lth trajectory to be non-salient.
3: Initialize $SM_l = -1$ and $Lsub_l = -1$;
4: If $k_l - j_l \leq 10$, continue;
5: Initialize counters: $P_u = P_v = N_u = N_v = 0$;
6: Counting consistency of horizontal flow direction u:
for $t = j_l$; $t <= k_l$; t++
if $\big(u_t(X_{l,t}) > 0.01\big)\, P_u = P_u + 1$;
if $\big(u_t(X_{l,t}) < -0.01\big)\, N_u = N_u + 1$;
end for
7: Repeating previous operation on the vertical flow component v, and obtain P_v, N_v;
8: **Condition 1:** if any one of P_u, P_v, N_u or N_v is greater than $0.8(k_l - j_l)$, $SM_l = max\ d_E(X_{l,t_1}, X_{l,t_2})$, for all $(t_1, t_2 \in [j_l, k_l])$, here d_E denotes Euclidean distance;
9: **Condition 2:** if Con.1 fails, partition the trajectory as described in Fig. 2. Then find the sub-trajectories P_aP_b with maximum length, $maxL = d_E(P_a, P_b)$; if the duration of P_aP_b is longer than 3 sec, $Lsub_l = maxL$.
10: **end for**
11: Obtain the maximum and minimum SM_{max}, SM_{min}, $Lsub_{max}$, $Lsub_{min}$ among all $SM_l > 0$ and $Lsub_l > 0$ $(l = 1, \ldots, m)$ respectively;
12: Normalize measures obtained across the two conditions: for those trajectories from Con.2 with $Lsub_l > 0$, $SM_l = SM_{min} + \frac{Lsub_l - Lsub_{min}}{Lsub_{max} - Lsub_{min}}(SM_{max} - SM_{min})$; in the unlikely event that there is no other trajectories declared salient from Con.1, define $SM_l = Lsub_l$.

3.4 Second-Pass RPCA

Based on the motion saliency computed in the preceding step, trajectories marked as non-salient ($SM_l = -1$) in Algorithm 2 are rejected. Now, with most of the non-stationary background motions filtered off, we can afford to lower λ in the second-pass RPCA step. This would ensure that all the changes caused by the foreground motion will be entirely transferred to the outlier matrix and not leave any ghostly presence in the background, yet without incurring a large false positive rate. Thus, for all blocks, we lower λ by at least one order of magnitude compared to before. For all blocks i with salient motion, we set $\lambda_i = \frac{0.1}{\sqrt{max(m,n)}} \frac{\overline{SM}_{min}}{\overline{SM}_i}$ where the last factor normalizes the $\overline{SM}_i$ computed for this block with respect to the minimum $\overline{SM}$ detected among all blocks containing salient motions. For blocks with no salient motion, the λ_i's are set to arbitrarily large values.

With the location and size of the likely outlier blocks estimated, and each weighted by a different saliency measure (Fig. 3), we are ready to carry out the second pass RPCA,

$$\min_{\boldsymbol{L},\boldsymbol{S}} \ ||\boldsymbol{L}||_* + \sum_i \lambda_i ||\boldsymbol{P}_i(\boldsymbol{S})||_F \quad \text{s.t. } \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}, \tag{5}$$

where the second term sums over all salient blocks, $||\cdot||_F$ is the Frobenius norm of a matrix, and $\boldsymbol{P}_i$ is an operator that unstacks each column of $\boldsymbol{S}$, and returns a matrix that represents block i. Essentially, this is the block-sparse version of the conventional RPCA that favors spatially contiguous outliers. Equation (5) remains a convex optimization problem and we solve it via the inexact ALM method. Interested readers can refer to Lin et al. [19] for details of this method. Briefly, the augmented Lagrangian function is defined as:

$$\begin{aligned} f(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{Y}, \mu) = ||\boldsymbol{L}||_* + \sum_i \lambda_i ||\boldsymbol{P}_i(\boldsymbol{S})||_F \\ + \langle \boldsymbol{Y}, \boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S} \rangle + \frac{\mu}{2} ||\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S}||_F^2, \end{aligned} \tag{6}$$

where $\boldsymbol{Y}$ is the Lagrange multiplier, μ is a positive scalar. For this block-sparse RPCA, besides the usual soft-thresholding operator $S_{\epsilon_i}[\cdot]$ needed for the minimization with respect to $\boldsymbol{L}$, we need the following block shrinkage (BS) operator during the minimization with respect to $\boldsymbol{S}$:

$$BS_{\epsilon_i}[\boldsymbol{G}_i] = \begin{cases} \frac{||\boldsymbol{G}_i||_F - \epsilon_i}{||\boldsymbol{G}_i||_F} \boldsymbol{G}_i & \text{if } ||\boldsymbol{G}_i||_F > \epsilon_i, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where $\boldsymbol{G}_i$ is a matrix representing the block i, and $\epsilon_i = \lambda_i \mu_k^{-1}$ is the corresponding threshold for this block. This shrinkage operator taken over all blocks has been proven to be the closed-form solution of the minimization with respect to $\boldsymbol{S}$ [31]. We summarize the algorithm in Algorithm 3.

Algorithm 3 Block-sparse RPCA via Inexact ALM.

Input: Matrix $\boldsymbol{M} \in \mathbb{R}^{m \times n}$, all salient blocks b_i, and the corresponding λ_i;
Output: Estimate of $(\boldsymbol{L}, \boldsymbol{S})$;
1: Initializing: $\boldsymbol{Y}_0 = \boldsymbol{M}/J(\boldsymbol{M})$, $\boldsymbol{S}_0 = 0$, $\mu_0 > 0$, $\rho > 1$, $k = 0$;
2: **while** not converged **do**
3: //Lines 4-5 solve $\boldsymbol{L}_{k+1}$=arg $\min_{\boldsymbol{L}}\ f(\boldsymbol{L}, \boldsymbol{S}_k, \boldsymbol{Y}_k, \mu_k)$, as equation (6).
4: $(\boldsymbol{U}, \boldsymbol{\Lambda}, \boldsymbol{V})$=svd$(\boldsymbol{M} - \boldsymbol{S}_k + \mu_k^{-1}\boldsymbol{Y}_k)$;
5: $\boldsymbol{L}_{k+1} = \boldsymbol{U} S_{\mu_k^{-1}}[\boldsymbol{\Lambda}] \boldsymbol{V}^T$.
6: //Line 7 solves $\boldsymbol{S}_{k+1}$=arg $\min_{\boldsymbol{S}}\ f(\boldsymbol{L}_{k+1}, \boldsymbol{S}, \boldsymbol{Y}_k, \mu_k)$.
7: Block-wise shrinkage: $\boldsymbol{S}_{k+1} = BS_{\lambda_i \mu_k^{-1}}[\boldsymbol{M} - \boldsymbol{L}_{k+1} + \mu_k^{-1}\boldsymbol{Y}_k]$;
8: $\boldsymbol{Y}_{k+1} = \boldsymbol{Y}_k + \mu_k(\boldsymbol{M} - \boldsymbol{L}_{k+1} - \boldsymbol{S}_{k+1})$, $\mu_{k+1} = min(\rho\mu_k, 10^7\mu_0)$.
9: $k \leftarrow k + 1$.
10: **end while**
11: Output $(\boldsymbol{L}_k, \boldsymbol{S}_k)$

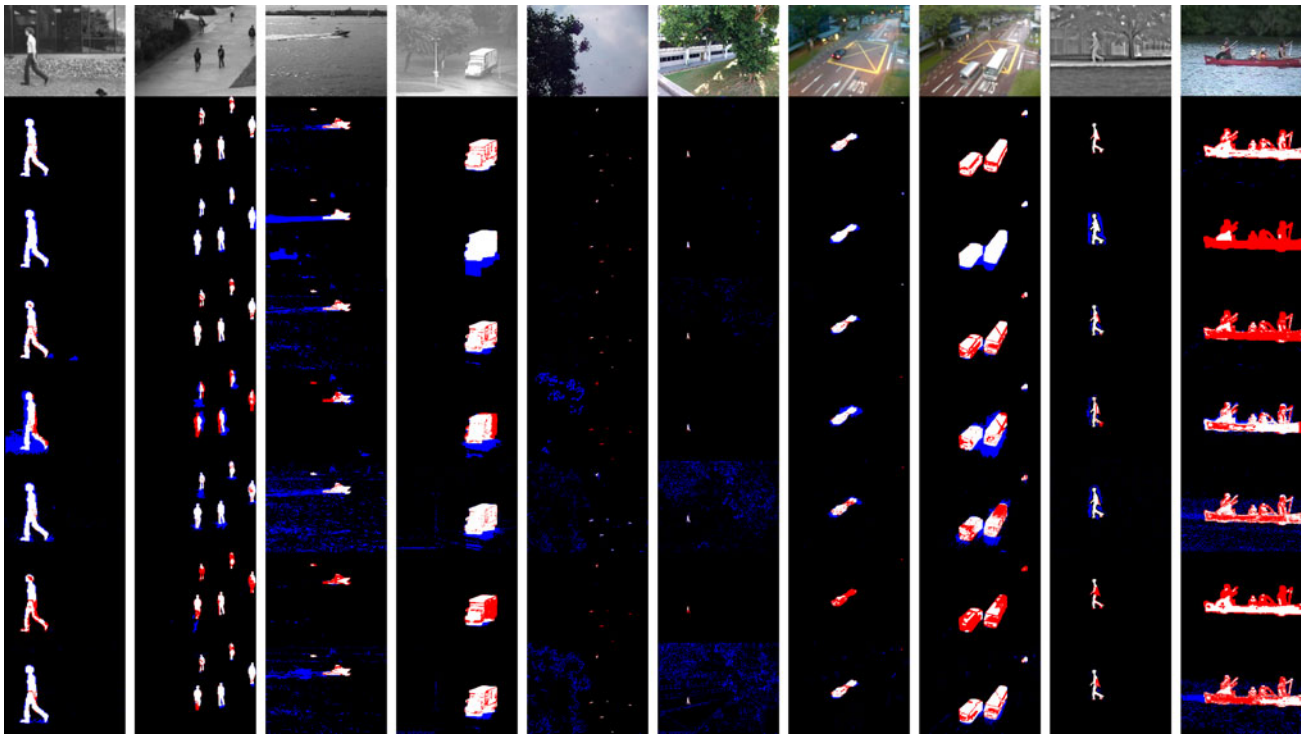


Fig. 4. Detected foreground mask of ten sequences depicted in each column. See online version for details. One image frame of the sequence is shown in the top row. Rows 2 to 8 depict the results of our **2-pass RPCA** method, DECOLOR, PCP, KDE, Zivkovic, Maddalena, Barnich respectively. For better visualization, we manually tagged the foreground in these frames and adopt such color coding scheme for the remainder of this paper: white represents correctly detected foreground, red missing pixels, and blue false alarm.

In Algorithm 3, we adopt the default values and conditions recommended by [19]. Specifically, $\mu_0 = 1.25/||\boldsymbol{M}||_2$, $\rho = 1.6$ and $J(\boldsymbol{M}) = max(||\boldsymbol{M}||_2, \lambda^{-1}||\boldsymbol{M}||_\infty)$, whereby $||\cdot||_2$ and $||\cdot||_\infty$ are the spectral norm and the l_∞ matrix norm respectively, and λ is set at $0.1/\sqrt{max(m,n)}$. The criteria for convergence at step 2 is $||M - \boldsymbol{L}_k - \boldsymbol{S}_k||_F/||\boldsymbol{M}||_F \ < \ 10^{-7}$.

4 Experiments and Analysis

We now perform experiments on both real and synthetic sequences. We first make a qualitative assessment of various methods by running them on a few specially chosen sequences and presenting the results at a few particular frames. We then make use of two recently released data sets, SABS [2] and CDnet [13], for a thorough quantitative assessment. We adopt the respective test criteria, which are *recall* and *precision* in SABS, and *F-measure* in CDnet:

$$\begin{aligned} recall &= \frac{\text{correctly classified foreground}}{\text{foreground in ground truth}}, \\ precision &= \frac{\text{correctly classified foreground}}{\text{pixels classified as foreground}}, \\ F\text{-}measure &= 2\frac{precision \cdot recall}{precision + recall}. \end{aligned}$$

In all these comparisons, our algorithm is denoted by the name *2-pass RPCA*.

In both the qualitative assessment (Section 4.1) and the SABS data set evaluation (Section 4.2.1), we compare our algorithm with the three best performing methods evaluated in [2], namely Zivkovic and van der Heijden [38], Barnich and Droogenbroeck [1], Maddalena and Petrosino [20], following the naming convention of Brutzer et al. [2].[2] We augment the above list with the Kernel Density Estimate method [9], as it is proposed specifically to deal with complex dynamic scenes but has not been included in [2]. We also compare our algorithm with two RPCA-based methods as these methods are closest to ours in spirit: the PCP of [4], and the DECOLOR of [37] which combines RPCA and MRF, and is claimed to outperform PCP.[3] The difference in performance between PCP and ours is essentially the performance gain brought about by the second pass with its block-specific λ setting based on motion saliency values. We would also have liked to include in our comparison the three best performing algorithms in [13]. One of them, the DECOLOR algorithm, is already included in the above; in the case of the other two, DPGMM [14] and SGMM-SOD [10], we are prevented from a comprehensive evaluation due to problem with codes availability and the requirement that clips must have clean background for training. In the case of DPGMM, the authors have kindly helped us run

2. These are known respectively as ViBe for Barnich, and SOBS for Maddalena in the terminology of the second data set CDnet.

3. Note that for the PCP and our method, a thresholding step is required to produce the final foreground mask, as many entries in $\boldsymbol{S}$ may contain vanishingly small values. To obtain a threshold, we first identify the likely outlier locations. Those pixels whose corresponding entries in $\boldsymbol{S}$ have magnitudes less than half of the maximum entries in $\boldsymbol{S}$ are regarded as background. Next obtain the difference between $\boldsymbol{M}$ and $\boldsymbol{L}$ at those tentatively identified background locations to estimate the expected level of noise. Finally, we set the threshold at the mean of the difference values plus three standard deviations of those difference values and apply it to $\boldsymbol{S}$.

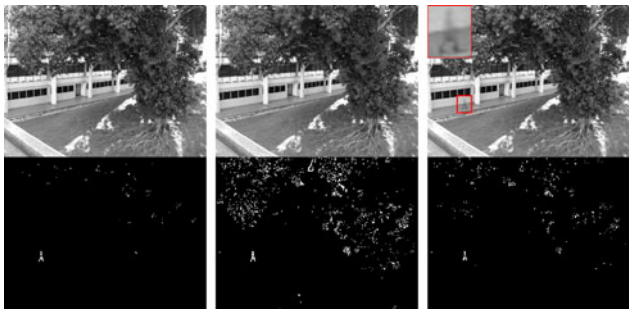


Fig. 5. Recovered background and foreground (top and bottom rows respectively). Left: our **2-pass RPCA** method; middle: PCP with $\lambda = 1/\sqrt{max(m,n)}$; right: PCP with $\lambda = 2/\sqrt{max(m,n)}$. Top right, inset: ghostly presence left in the background.

some tests on clips where relatively clean background is available, and the results are presented separately in Section 4.1.

4.1 Qualitative Results

Fig. 4 shows the detected foreground masks on 10 real image sequences, the first four from various public data sets used in previous works, followed by four collected by us to demonstrate specific issues addressed by our research and lastly, two clips from the CDnet data set. Readers are advised to refer to the online version for details in the figures.

The first two columns in Fig. 4 (people walking) are basically the baseline "easy" cases, where most algorithms performed well, though with different degrees of details in the foreground detected. The third column depicts a boat scene with wave motion. This presents difficulties for many algorithms; only ours and Maddalena performed relatively well. Note that both the large speedboat in the middle and the small speedboat in the far distance are detected by our method, demonstrating its scale-invariance. The fourth and the eighth columns depict rainy scenes where falling rain causes "motion" in the background. Evaluated over the two scenes, our method again produced the best results. Maddalena is able to remove the background clutter very well, but it tends to produce incomplete foreground object, as well as missing small moving object (eighth column).

The fifth column depicts a swaying tree scene with various small birds flying. The three best performing methods in [2], namely, Zivkovic, Barnich, and Maddalena, all failed to do well. Either they picked up the tree motions, or failed to detect the birds. The KDE method that is introduced to handle dynamic background also failed quite badly in this sequence. The RPCA-based methods did better, with our method correctly picking out all the birds as foreground without any of the tree motion. These positive results also showed that our motion saliency measure can handle zigzagging manoeuvers. Most such evasive manoeuvers have an overall direction, that is, they result in a net movement in certain directions, thus meeting the criteria for motion saliency. The sixth column depicts the campus scene discussed before, with a large swaying tree and a walking person. Due to its proximity to the camera, the magnitude of the tree motion is larger than that of the human. As can be seen from the results of our method, the human silhouette is cleanly delineated. For other methods, only the KDE and Maddalena did not have any false alarms, but the human silhouettes were not as cleanly detected. The problems faced by other RPCA-based method will be further commented upon in Fig. 5.

The seventh column depicts an evening scene with high image noise and flickering illumination caused by fluorescent lights. The sudden change in intensity caused by the varying illumination is not a problem for the RPCA-based methods, as the effect of the change is entirely captured by the low rank constraint. The KDE and Maddalena can also handle the sudden change in illumination well, but both of them missed the small car at the top right corner of the image. In both the seventh and the eighth columns, our method can also detect small camouflaged cars moving in the carpark behind the row of roadside trees (not present in the frame shown).

The ninth and 10th columns are from the CDnet data set. The ninth column depicts a thermal sequence captured by a far-infrared camera. Typical thermal artifacts exist, such as heat emission which results in an apparent target that is larger than its true size. As can be seen, our method performed best with little missed detection along the foreground boundary. DECOLOR and Zivkovic achieved complete foreground detection, but with much more false alarm and distinct oversmoothing. Lastly, the 10th column depicts a canoe scene with mild wave motion and rowing action. DECOLOR and PCP achieved a rather low recall; a likely explanation is that the sparse outlier assumption is violated here with the large foreground object size. In contrast, via adaptively setting the regularizing parameter, our method can handle object of large size, obtaining much more complete foreground detection.

From the above, the qualitative conclusions that we can draw are: (1) despite the claim made in [2] that the best performing algorithms can handle dynamic background, this is not true when the background motion is large enough, as can be seen from some of the cases tested here, and also by the quantitative results obtained on the CDnet data set (see Table 2 with its rapidly decreasing F-measures); (2) While KDE and Maddalena can handle dynamic background well, they fail in some cases, and the foreground objects returned are often incorrect in shape, incomplete, or missed altogether (especially if the objects are small); (3) RPCA-based methods can handle various background changes quite well generally; these include illumination changes, changes caused by rain, tree and wave motions. Our method and PCP tend to produce cleanly delineated foreground shapes; this is typically not the case for DECOLOR due to the MRF smoothness prior imposed on the foreground shapes. Both PCP and DECOLOR fail when the foreground object is too large.

Despite the general success of the RPCA-based methods, PCP and to a lesser extent, DECOLOR have problems in setting a correct regularizing parameter that can handle regions or motions of varying scales, which is what we set out to overcome. We now explore this point in greater depth in Fig. 5. It can be seen that no matter how the value of λ is chosen, we cannot obtain a simultaneously satisfactory background and foreground for PCP. On the one hand, when λ is small (middle column), no ghost of the foreground is detected in the recovered background but unfortunately,

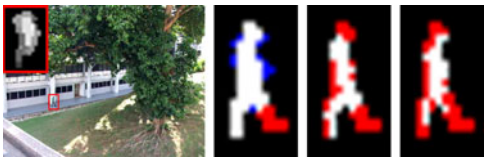


Fig. 6. Combining one-pass RPCA with motion saliency filtering. From left to right: saliency of the human figure with brighter intensity denoting higher saliency values; results of foreground detection, with $\lambda = 0.5, 1$, and 2 respectively.

much clutter remains in the foreground. On the other hand, setting a somewhat larger λ (right column) has the undesirable effect of putting some of the genuine foreground in the background, even when the foreground itself still contains many false alarm. When viewed in video sequence, the effect of a ghostly presence walking in the background when the human is not cleanly removed is much clearer. For our results, there is no ghost and the motion of the swaying tree is largely retained in the background, creating a pleasing video edit that has little artifacts.

All the results mentioned above show that the spatial contiguity prior that we incorporate in the form of blocks and the motion saliency measure have been quite effective in handling the aforementioned issues. Unlike DECOLOR which enforces the MRF constraint, our method is sensitive to small targets and details; it does not suffer from merging of adjacent objects nor inflation of foreground area.

We have also shown in Fig. 6 the campus scene results produced by an one-pass RPCA (i.e., PCP), coupled with the motion saliency step as a post-processing filter that removes non-salient regions. Three typical values of λ are used, but all results are inferior compared to that of our two-pass RPCA (top left of Fig. 7). The reasons for the poor performance are twofold: first, such strategy is limited to using a global λ rather than a block-specific λ; second, the practical difficulties of optical flow estimation (e.g., over-smoothing at motion boundaries) means that the motion saliency values computed are not accurate enough for precise foreground separation.

Finally, many better algorithms from the CDnet evaluation [13] are GMM-based, and thus require clips with clean background for training. We now present the results

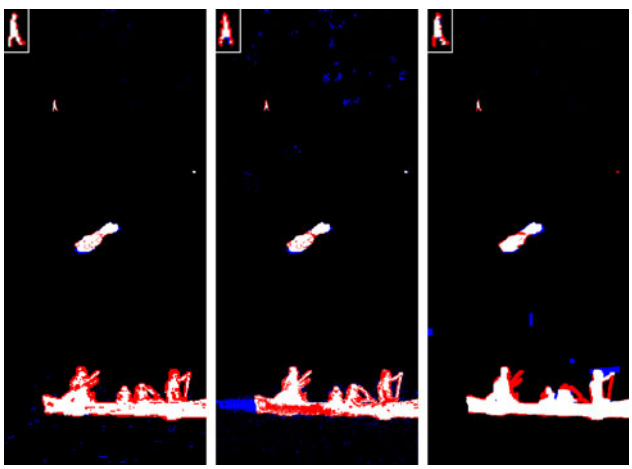


Fig. 7. Detected foreground masks of *2-pass RPCA* method (left), ReProCS (middle) and DPGMM (right) on seq.6, 7, 10 in Fig. 4.

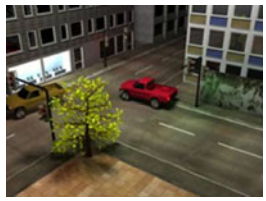


Fig. 8. Synthetic scene of the SABS data set.

obtained by DPGMM (the best algorithm from the CDnet evaluation at the time of publication) on sequences 6, 7, and 10, where such clean backgrounds are available. We also take this chance to compare against those methods that estimate the sparse outlier support automatically, in particular, ReProCS [24] which also requires sequences with clean background. As shown in Fig. 7, in sequences 6 and 7, where the amount of training data is only 50 frames out of a total of 100 frames, DPGMM performed not as well when compared with our method, missing some of the human foreground in sequence 6 (see inset for details) and a small car entirely in sequence 7. In sequence 10, where long training data is available (more than 500 frames), DPGMM performed quite well, obtaining less missed detection but more false alarm compared to our method. Thus it seems that significant amount of training frames must be available in order for DPGMM to achieve its excellent best performance. When compared to ReProCS, our method performed significantly better in sequences 6 and 10 and slightly better in sequence 7. These results corroborate our previous claim that our explicit two-pass method can better handle foreground detection amidst complex background motion.

4.2 Quantitative Results

4.2.1 SABS Data Set

The SABS data set comprise synthetic image sequences divided into nine test categories, with challenging scenarios such as gradual and sudden illumination changes, camouflage, dynamic background, etc. While the images are not real (see Fig. 8), the data set allows a controlled testing of various challenges, some of which are not available in the CDnet data set (e.g., sudden illumination change, high noise in dim condition, and camouflage). Out of the nine test categories, we only include the following four for comparison: *Light Switch* (sudden illumination change), *Noisy Night* (dim ambient lighting), *dynamic background* and *camouflage*. Other categories are either very easy for most techniques or peripheral to our concern here and are hence not included.

Fig. 9 shows the precision-recall charts of the performance of different methods with varying thresholds. It is obvious that our method significantly outperforms other methods evaluated in [2], reaching more than 70 percent precision at recall higher than 90 percent. It is worth pointing out that the dynamic background motion created in the synthetic SABS data set is really quite small in magnitude; otherwise, the superiority of our method in this case will be even more conspicuous. The relatively good performance reported in [2] is clearly not borne out by the results shown in our preceding qualitative experiments on real sequences, as well as the results from CDnet, with their larger and more realistic motion. The RPCA-based methods generally

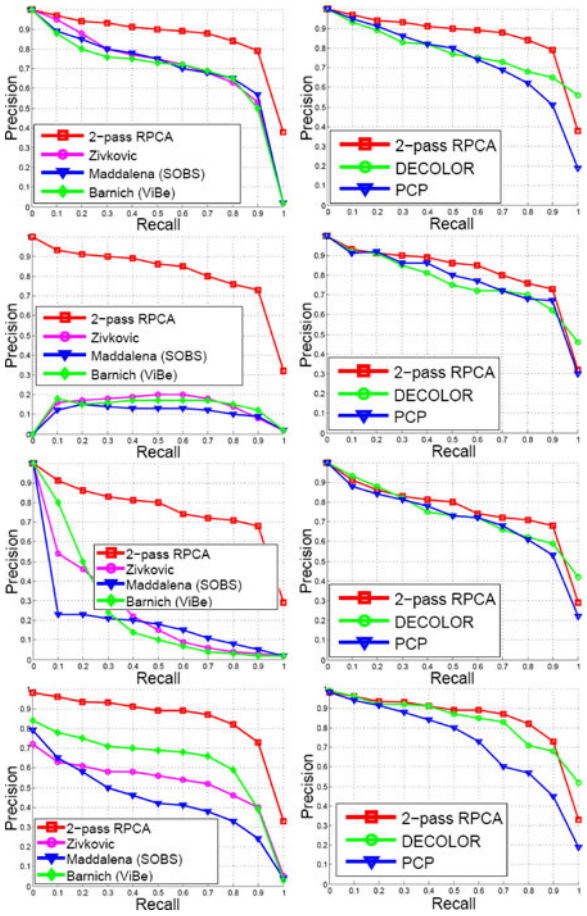


Fig. 9. Precision-recall charts of different methods with varying threshold. first row: Dynamic Background; second row: Light Switch; third row: Noisy Night; fourth row: Camouflage.

perform better than the others; among these RPCA-based methods, our method is the clear winner. However, at very high recall rates (about 95 percent and above), DECOLOR performs better than ours. This is due to the fact that the MRF-based smoothness prior of DECOLOR ensures that, as the threshold is varied, additional foreground points only grow around the core foreground regions already detected. This feature ensures that the additional points detected are not some background clutter; it is indeed a virtue against which we cannot prevail unless we are willing to forego the scale-independent detection ability of our algorithm. That is, if we accept that once beyond certain threshold, additional foreground points can only come in the blocks where currently detected foreground regions reside, we are essentially giving up on any remaining small, inconspicuous foreground objects that are still not detected. This can readily be done, if specific application needs so dictate.

4.2.2 CDnet Data Set

The second data set CDnet consists of 31 real-world videos (including thermal sequences) totalling over 80,000 frames and spanning six categories selected to include diverse motion and change detection challenges (see Fig. 10). For each category, we compare our two-pass RPCA algorithm with the top-performing methods which have submitted results for that category for the reason of space limit (readers are referred to Goyette et al. [13] and its website for the

Fig. 10. Representative scenes of the six categories in CDnet data set. From top, left to right: *Baseline*, *Dynamic Background*, *Shadow*, *Thermal*, *Intermittent Object Motion*, *Camera Jitter*.

complete list of references and the corresponding performance figures). The addition we made to this rather comprehensive list includes DECOLOR and PCP. In DECOLOR, there is an explicit alignment operation to handle camera jitter; for PCP, we augment the algorithm with the alignment step presented in Section 3.2, and refer to it as PCP+Alignment in the following tables. For ease of comparison, we adhere to the definition of background set forth in this data set when evaluating the algorithms' performance, even though some of these definitions might be arguable. For instance, the accompanying shadow of a moving foreground is considered as background in the ground truth, while our algorithm considers it as part of the foreground, which means that our algorithm's performance will be negatively affected by this definition. Another point to note is that the CDnet data set defines a ROI (region of interest) for each sequence; only the pixels contained in the ROI are counted towards the performance metric. Furthermore, the ROI usually contains only objects at the same scale. As a result, there is no test of algorithms' ability to pick up foreground objects of different scales.

Table 1 shows the results of the *Baseline* category, which comprises four fairly easy videos. Clearly, all algorithms perform well, reaching a F-measure of more than 0.9.[4]

Table 2 shows the results of the *Dynamic Background* category, comprising six videos depicting outdoor scenes with strong background motion, two with boats on shimmering water, two with cars passing near a fountain, and the last two with pedestrians, cars and trucks passing in front of a scrub swaying under wind. Compared with Table 1, this challenge clearly presents difficulties for all algorithms, resulting in a drop of F-measure by about 10 percent in the top performing algorithms. It is also interesting to compare the performance of PSP-MRF [26] and DECOLOR. Both use the MRF constraint, and one would expect that they tend to over-smooth the foreground region, thus resulting in high recall rate. However, DECOLOR achieved a rather low recall; a visual inspection of the results seems to indicate that the DECOLOR

4. Note that for layout purposes, we abbreviate the names of the following methods:
Chebyshev 1 : Chebyshev probability approach
Chebyshev 2 : Chebyshev prob. with static object detection
QCHBMD : Quasi-continuous histograms based motion detection
KDE-STCD : KDE-spatio-temporal change detection
KDE-ISTF : KDE-integrated spatio-temporal features
LSS : Local-self similarity.

TABLE 1
Results of the *Baseline* Sequences

method	F-measure	recall	precision
SC-SOBS	**0.9333**	0.9327	0.9341
PSP-MRF	0.9289	0.9319	0.9261
DPGMM	0.9286	0.9632	0.8984
2-pass RPCA	0.9281	0.9261	0.9301
SOBS	0.9251	0.9193	0.9313
PBAS	0.9242	0.9594	0.8941
SGMM-SOD	0.9223	0.9407	0.9072
DECOLOR	0.9215	0.9306	0.9126
CDPS	0.9208	0.9488	0.8969
PCP+Alignment	0.9109	0.9238	0.8984
KDE-Elgammal	0.9092	0.8969	0.9223
Histogram	0.9004	0.8777	0.9254
M-distance	0.8954	0.8872	0.9071
E-distance	0.8720	0.8385	0.9114
ViBe+	0.8715	0.8283	0.9262
ViBe	0.8700	0.8204	0.9288
Chebyshev 2	0.8646	0.8266	0.9143
SGMM	0.8594	0.8680	0.8584
LSS	0.8494	0.9732	0.7564
KNN	0.8411	0.7934	0.9245

framework cannot handle foreground objects with large sizes. This echoes the findings in the preceding section, and corroborates what we said earlier about the problem of scale and the difficulty of setting the value of the regularizing parameter λ. Finally, we note that when there are enough clean background data for training, DPGMM is very competitive, turning in the best performance in this category, as well as in the *Thermal* category later.

Table 3 shows the results of the *Shadows* category, which consists of six videos (two indoor and four outdoor) exhibiting shadows with different intensities and sizes. Our method ranks ninth, the worst ranking among all categories. However, due to the simple scene content in this data category, the F-measure of our method is about 0.8, still better than those of many other categories. It should be noted that among the methods ahead of ours, Chebyshev 1 and 2 have a specific shadow processing step. SGMM-SOD, the best performing algorithm in this category, maintained two background models with different learning rates, and therefore is better able to handle short-term phenomena, like shadow in the background here, as well as intermittent stops of the foreground in Table 5.

TABLE 3
Results of the *Shadow* Sequences

method	F-measure	recall	precision
SGMM-SOD	**0.8613**	0.9184	0.8187
PBAS	0.8597	0.9133	0.8143
Chebyshev 1	0.8333	0.8669	0.8104
Chebyshev 2	0.8333	0.8670	0.8103
DECOLOR	0.8317	0.8816	0.7871
ViBe+	0.8153	0.8108	0.8302
DPGMM	0.8127	0.8545	0.8240
CDPS	0.8092	0.9233	0.7567
2-pass RPCA	0.8063	0.8627	0.7567
ViBe	0.8032	0.7833	0.8342
KDE-Elgammal	0.8028	0.8536	0.7660
SGMM	0.7944	0.8580	0.7617
PSP-MRF	0.7907	0.8736	0.7281
PCP+Alignment	0.7885	0.8416	0.7417
SC-SOBS	0.7786	0.8502	0.7230
SOBS	0.7716	0.8350	0.7219
KDE-ISTF	0.7545	0.7197	0.8244
KNN	0.7468	0.7478	0.7788
GMM-Stauffer	0.7370	0.7960	0.7156
GMM-Rectgaus	0.7331	0.7189	0.7840

Table 4 shows the results of the *Thermal* category, which comprises five videos (three outdoor and two indoor) captured by far-infrared cameras. Although the type of change or the motion is simple, the resulting performance is still much worse than that of the *Baseline* due to a variety of typical thermal artifacts, such as heat emission which results in an apparent target that is larger than its true size, heat stamps (e.g., bright spots left on a seat after a person gets up and leaves), heat reflection on floors and windows, and camouflage effects arising from a foreground target having the same temperature as the surrounding regions. Here the oversmoothing problem of DECOLOR is especially apparent (see Fig. 4, second last column, third row).

Table 5 shows the results of the *Intermittent Object Motion* category, which comprises six videos depicting objects with stop and start motions. With continuous tracking even when the object has stopped, our method has no problem in handling intermittent motions. Nevertheless, we would like to add the caveat that if the foreground object stops long enough, then it is arguable whether one should continue to treat this as foreground or as part of the background. We are not able to comment on the best performing algorithm

TABLE 2
Results of the *Dynamic Background* Sequences

method	F-measure	recall	precision
DPGMM	**0.8137**	0.8852	0.7762
2-pass RPCA	0.7818	0.8392	0.7368
Chebyshev 1	0.7656	0.8182	0.7633
Chebyshev 2	0.7520	0.8182	0.7339
CDPS	0.7495	0.7590	0.8086
ViBe+	0.7197	0.7616	0.7291
DECOLOR	0.7084	0.6682	0.7538
PSP-MRF	0.6960	0.8955	0.6576
PCP+Alignment	0.6941	0.7236	0.6669
KNN	0.6865	0.8047	0.6931
PBAS	0.6829	0.6955	0.8326
SGMM-SOD	0.6826	0.7715	0.7198
GMM-Kaew	0.6697	0.6303	0.7700
SC-SOBS	0.6686	0.8918	0.6283
KDE-STCD	0.6574	0.8935	0.5888
SOBS	0.6439	0.8798	0.5856
QCHBMD	0.6430	0.8909	0.5347
SGMM	0.6380	0.7715	0.6665
GMM-Stauffer	0.6330	0.8344	0.5989
GMM-Zivkovic	0.6328	0.8019	0.6213

TABLE 4
Results of the *Thermal* Sequences

method	F-measure	recall	precision
DPGMM	**0.8134**	0.8869	0.7629
2-pass RPCA	0.7597	0.7125	0.8136
PBAS	0.7556	0.7283	0.8922
KDE-Elgammal	0.7423	0.6725	0.8974
LSS	0.7297	0.9036	0.6433
UBA	0.7283	0.6880	0.7962
Chebyshev 1	0.7259	0.6940	0.8910
Chebyshev 2	0.7230	0.6887	0.8906
PCP+Alignment	0.7192	0.6613	0.7882
DECOLOR	0.7081	0.8268	0.6192
SGMM-SOD	0.7081	0.6089	0.9515
M-distance	0.7065	0.6270	0.8617
Histogram	0.6996	0.6412	0.8110
Bayesian BG	0.6969	0.6026	0.8877
PSP-MRF	0.6932	0.5991	0.9218
SC-SOBS	0.6923	0.6003	0.8857
SOBS	0.6834	0.5888	0.8754
ViBe	0.6647	0.5435	0.9363
ViBe+	0.6646	0.5411	0.9477
GMM-Stauffer	0.6621	0.5691	0.8652

TABLE 5
Results of the *Intermittent Object Motion* Sequences

method	F-measure	recall	precision
CDPS	**0.7406**	0.8084	0.7624
SGMM-SOD	0.6957	0.7305	0.7834
UBA	0.6886	0.7205	0.7310
2-pass RPCA	0.6826	0.6528	0.7152
DECOLOR	0.5945	0.7262	0.5032
SC-SOBS	0.5918	0.7237	0.5896
PBAS	0.5745	0.6700	0.7045
PSP-MRF	0.5645	0.7010	0.5727
SOBS	0.5628	0.7057	0.5531
KDE-ISTF	0.5454	0.4512	0.8166
DPGMM	0.5418	0.6763	0.6525
SGMM	0.5397	0.5013	0.6993
PCP+Alignment	0.5371	0.4315	0.7113
LSS	0.5329	0.9027	0.4445
GMM-Zivkovic	0.5325	0.5467	0.6458
GMM-Stauffer	0.5207	0.5142	0.6688
Histogram	0.5112	0.7512	0.4859
ViBe+	0.5093	0.4729	0.7513
ViBe	0.5074	0.5122	0.6515
KDE-STCD	0.5039	0.4372	0.7212

TABLE 6
Results of the *Camera Jitter* Sequences

method	F-measure	recall	precision
2-pass RPCA	**0.8152**	0.7978	0.8335
DECOLOR	0.7776	0.7721	0.7832
ViBe+	0.7538	0.7293	0.8064
PSP-MRF	0.7502	0.8211	0.7009
DPGMM	0.7477	0.6988	0.8426
SGMM	0.7251	0.7088	0.7752
PBAS	0.7220	0.7373	0.7586
PCP+Alignment	0.7218	0.6264	0.8515
KDE-STCD	0.7122	0.7562	0.6793
KDE-ISTF	0.7110	0.7316	0.6993
SOBS	0.7086	0.8007	0.6399
SC-SOBS	0.7051	0.8113	0.6286
SGMM-SOD	0.6988	0.6310	0.8273
KNN	0.6894	0.7351	0.7018
Chebyshev 2	0.6416	0.7223	0.5960
ViBe	0.5995	0.7112	0.5289
Bayesian BG	0.5988	0.5441	0.6678
GMM-Stauffer	0.5969	0.7334	0.5126
GMM-Kaew	0.5761	0.5074	0.6897
KDE-Elgammal	0.5720	0.7375	0.4862

CDPS as its authorship remains anonymous at the time of publication.

Table 6 shows the results of the *Camera Jitter* category. Due to the respective explicit alignment step, both our method and DECOLOR perform well. Some of the other non-RPCA based methods may also derive certain degree of robustness to camera jitters via their stochastic non-parametric background modeling strategy (e.g., ViBe+ and PBAS, etc.).

We also show the detected foreground masks of the *Traffic* and the *Sidewalk* sequences in Figs. 11 and 12 to depict the effectiveness of our image alignment step in comparison to those of DECOLOR and PCP. The *Traffic* sequence contains the largest jitter displacement (up to about 20 pixels in the original resolution), and the *Sidewalk* sequence has mild jitter but the smallest target. See Table 8 for the jitter displacements estimated in the scaled-down resolution. Both our method and DECOLOR can handle the large pixel displacement in the *Traffic* sequence. On the *Sidewalk* sequence, neither DECOLOR nor PCP can detect the lower limbs and the shadow of the person which only moved a few pixels in

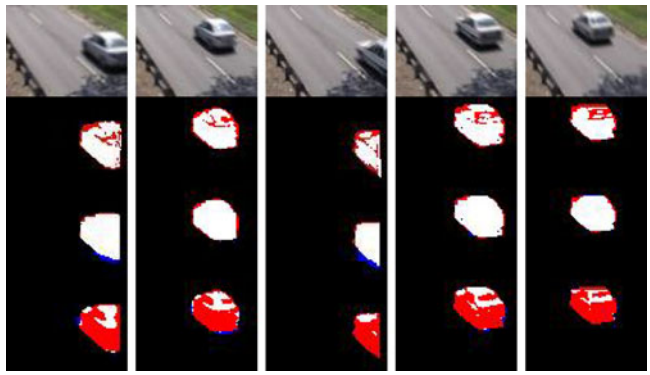


Fig. 11. Detected foreground mask of the *Traffic* sequence. The original image frames are shown in the top row. Rows 2 to 4 depict the results of our two-pass RPCA method, DECOLOR plus alignment, PCP plus alignment respectively.

Fig. 12. Detected foreground mask of the *Sidewalk* sequence.

TABLE 7
Overall Results over all Categories

method	F-measure	recall	precision
2-pass RPCA	**0.7956**	0.7985	0.7977
DPGMM	0.7763	0.8275	0.7928
SGMM-SOD	0.7624	0.7681	0.8351
DECOLOR	0.7570	0.8009	0.7265
PBAS	0.7532	0.7840	0.8160
PSP-MRF	0.7372	0.8037	0.7512
PCP+Alignment	0.7286	0.7014	0.7762
SC-SOBS	0.7283	0.8017	0.7315
CDPS	0.7281	0.7769	0.7610
ViBe+	0.7224	0.6907	0.8318
SOBS	0.7159	0.7882	0.7179
SGMM	0.7008	0.7073	0.7812
Chebyshev 2	0.7001	0.7133	0.7856
KNN	0.6785	0.6707	0.7882
KDE-Elgammal	0.6719	0.7442	0.6843
ViBe	0.6683	0.6821	0.7357
GMM-Stauffer	0.6624	0.7108	0.7012
GMM-Zivkovic	0.6596	0.6964	0.7079
KDE-STCD	0.6437	0.6576	0.7341
KDE-ISTF	0.6418	0.6507	0.7663

the sequence. Furthermore, we also observe that the performance of PCP is much more inferior to its performance over the static camera sequences. The root of this problem lies in that the recovered background component is not so low-rank due to imperfection in the alignment, with its rank value typically about two times higher than those of the static sequences. This increase in rank could now absorb some of the smaller, genuine foreground changes into the

TABLE 8
Results of Sequence Alignment

	Estimated jitter displacement (dx, dy) in scaled-down resolution.				
Seq.	frame 1	frame 20	frame 40	frame 60	frame 80
Traffic	1.66, 2.95	6.27, 6.30	-0.51, 1.33	3.83, 4.25	5.96, 4.82
Sidewalk	3.16, 2.54	1.72, -0.47	2.18, 3.73	1.93, 1.22	2.53, 3.77

(Unit: pixel).

background. This is precisely where the advantage of our adaptive weighing comes to the forth; it allows more of these weaker changes being decomposed as foreground via penalizing less with a smaller λ value. Note that this is less of an issue with the *Traffic* sequence, because while higher rank also exists in the recovered background, the foreground changes are typically of much larger magnitudes in this sequence such that there is no question of them being absorbed in the background.

Table 7 shows the overall ranking obtained across all six categories. Our algorithm emerges as the overall winner by virtue of its adaptability to various challenges, attaining near top performances in all categories rather than being first in particular challenges. To conclude, the quantitative assessment further corroborates the results obtained in the qualitative experiments: our algorithm works well under a wide variety of scenes and significantly outperforms several state-of-the-art techniques.

Lastly, we report on the amount of computations incurred by our two-pass RPCA algorithm on a Fujitsu Lifebook with dual quad-core 2.5 GHz Intel processors and 8 GB RAM executing Matlab codes. Excluding the optical flow estimation step, the average processing time on a sequence of 100 frames with resolution 320×240 is about 1,280 seconds. If the image alignment step is also performed, the average time increases to about 1,730 seconds.

5 Conclusion and Future Work

To handle the complex scenarios encountered in background subtraction work, we propose a hierarchical RPCA process which makes little specific assumption about the background. The two-pass RPCA process is interleaved with a motion saliency estimation step that makes our method yield substantially better results than conventional RPCA. We are able to incorporate the spatial contiguity prior in the form of blocks whose size and locations are detected automatically, without having to resort to smoothness prior such as MRF, thereby fully realizing the potential of the low rank representation method to return scale-independent, crisply defined foreground regions. Extensive experiments on challenging videos and benchmark data set demonstrate that our method outperforms various state-of-the-art approaches and works effectively on a wide range of complex scenarios.

Appendix

Point Tracking

Point $(x_t, y_t)^T$ can be tracked by using the flow field $\boldsymbol{w}(x, y) := (u(x, y), v(x, y))^T$ that $(x_{t+1}, y_{t+1})^T = (x_t, y_t)^T + (u_t(x_t, y_t), v_t(x_t, y_t))^T$. Since $(x_{t+1}, y_{t+1})^T$ usually ends up between grid points, we use bilinear interpolation to infer the flow. In new frame, we initialize new tracks at the points that no existing trajectories passed nearby (within 0.1 pixel unit).

We stop the tracking of a point as soon as it gets occluded. We detect occlusions by checking that the backward flow vector should point in the inverse direction as the forward flow: $w_t(x_t, y_t) = -\hat{w}_t(x_t + u_t, y_t + v_t)$, where $\hat{w}_t(x, y) := (\hat{u}_t(x, y), \hat{v}_t(x, y))$ is the flow from frame $t + 1$ to t. If the consistency is broken, the point is either being occluded or the flow was not correctly estimated. As small optical flow estimation errors are inevitable, we allow the following tolerance factor in the consistency check: $|\boldsymbol{w}(x, y) + \hat{\boldsymbol{w}}(x, y)|^2 < 0.01(|\boldsymbol{w}(x, y)|^2 + |\hat{\boldsymbol{w}}(x, y)|^2) + 0.2$.

For points on motion boundaries, the tracking may not be stable, with the undesirable result that the tracked points wander back and forth across the boundary. If not handled properly, such feature points might come to be regarded as undergoing inconsistent motion due to this drifting across the boundary. To ameliorate this effect, we perform two mitigating measures: First, if the tracking of a point near an edge experiences substantial variation in its motion, we at once stop the tracking and re-initialize a new feature track from this frame onwards. Specifically, the conditions for stopping the tracking are:

$$\begin{cases} |\nabla u|.^2 + |\nabla v|^2 > 0.01|\boldsymbol{w}|^2 + 0.002 \\ \sqrt{(\partial I/\partial x)^2 + (\partial I/\partial y)^2} > T_d, \end{cases}$$

where T_d is the average edge strength. Second, before the second pass RPCA, a border five pixels is added to each rectangular block so that no outlier pixels are missed. If intermittent object motions are prevalent in the scenes and it is desirable to continue regarding them as foreground even after they have stopped for a while, the user has an option to disregard the aforementioned stopping condition so that stationary points are continuously tracked, which is what we did in the *Intermittent Object Motion* category. Of course, this significantly increases the computational burden of the tracking process.

Acknowledgments

The authors would like to thank Tom SF Haines and Dr. Tao Xiang from University College London for running their DPGMM code on our sequences for experimental comparison. This work is supported by these Grants: JPP Grant R-263-000-A24-232, PSF Grant 1321202075, theory and methods of digital conservation for cultural heritage (2012CB725300), and the Singapore NRF under its IRC@SG Funding Initiative and administered by the IDMPO at the SeSaMe centre.

References

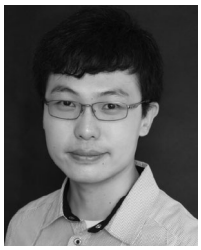
[1] O. Barnich and M. V. Droogenbroeck, "ViBE: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[2] S. Brutzer, B. Hoferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1937–1944.

[3] A. Bugeau and P. Perez, "Detection and segmentation of moving objects in complex scenes," *Comput. Vis. Image Understanding*, vol. 113, no. 4, pp. 459–476, 2009.
[4] E. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.
[5] J. Carreira and C. Sminchisescu, "CPMC: Automatic object segmentation using constrained parametric Min-Cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1312–1328, Jul. 2012.
[6] A. Cavallaro and T. Ebrahimi, "Video object extraction based on adaptive background and statistical change detection," in *Proc. SPIE, Vis. Commun. Image Process.*, 2001, pp. 465–475.
[7] V. Cevher, A. Sankaranarayanan, F. Duarte, D. Reddy, G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Proc. Eur. Conf. Comp. Vis.*, 2008, pp. 155–168.
[8] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3042–3054, Jun. 2010.
[9] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. 6th Eur. Conf. Comput. Vis.—Part II*, 2000, pp. 751–767.
[10] R. H. Evangelio, M. Pätzold, and T. Sikora, "Splitting Gaussians in mixture models," in *Proc. 9th IEEE Int. Conf. Adv. Video Signal-Based Surveillance*, 2012, pp. 300–305.
[11] A. Ganesh, J. Wright, X. Li, E. Candes, and Y. Ma, "Dense error correction for low-rank matrices via principal component pursuit," in *Proc. Int. Symp. Inf. Theory*, 2010, pp. 1513–1517.
[12] Z. Gao, L.-F. Cheong, and M. Shan, "Block-sparse RPCA for consistent foreground detection," in *Proc. 12th Eur. Conf. Comput. Vis.—Volume Part V*, 2012, pp. 690–703.
[13] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark data set," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog. Workshops*, 2012, pp. 1–8.
[14] T. Haines and T. Xiang, "Background subtraction with dirichlet processes," in *Proc. 12th Eur. Conf. Comput. Vis.—Volume Part IV*, 2012, pp. 99–113.
[15] J. Huang, X. Huang, and D. Metaxas, "Learning with dynamic group sparsity," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 64–71.
[16] S. Huwer and H. Niemann, "Adaptive change detection for real-time surveillance applications," in *Proc. 3rd IEEE Workshop Vis. Surveillance*, 2000, pp. 37–46.
[17] P. KadewTraKuPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Proc. 2nd Eur. Workshp Adv. Video-Based Surveil. Syst.*, 2011, pp. 135–144.
[18] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.
[19] Z. Lin, M. Chen, and Y. Ma., "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrix," in *Arxiv preprint arxiv.org/abs/1009.5055*, 2009.
[20] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1168–1177, Jul. 2008.
[21] T. Matsuyama, T. Ohya, and H. Habe, "Background subtraction for non-stationary scenes," in *Proc. 4th Asi. Conf. Comput. Vis*, 2000, pp. 662–667.
[22] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptative kernel density estimation in," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2004, pp. 302–309.
[23] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.
[24] C. Qiu and N. Vaswani, "ReProCS: A missing link between recursive robust PCA and recursive sparse recovery in large but correlated noise, " in Arxiv preprint arxiv.org/abs/1106.3286, 2011.
[25] K. Rosenblum, Z. Manor, and Y. Eldar, "Dictionary optimization for block-sparse representations," in *IEEE Trans. Signal Process*, vol. 60, no. 5, pp. 2386–2395, May. 2012.
[26] A. Schick, M. Bäuml, and R. Stiefelhagen, "Improving foreground segmentations with probabilistic superpixel markov random fields," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog. Workshops*, 2012, pp. 27–31.
[27] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov. 2005.
[28] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 1999, pp. 246–252.
[29] M. Stojnic, F. Parvaresh, and B. Hassibi, "On the reconstruction of block-sparse signals with an optimal number of measurements," *IEEE Trans. Signal Process.*, vol. 57, no. 8, pp. 3075–3085, Aug. 2009.
[30] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 2432–2439.
[31] G. Tang and A. Nehorai, "Robust principal component analysis based on low-rank and block-sparse matrix decomposition," in *Proc. 45th Annu. Conf. Inform. Sci. Syst.*, 2011, pp. 1–5.
[32] T. Veit, T. Cao, and P. Bouthemy, "A maximality principle applied to a contrario motion detection," in *Proc. IEEE Int. Conf. Image Process.*, 2005, pp. 1061–1064.
[33] L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, p. 774-780, Aug. 2000.
[34] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," in *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.
[35] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
[36] J. Zheng, Y. Wang, N. L. Nihan, and M. E. Hallenbeck, "Extracting roadway background image: Mode-based approach," *Transp. Res. Rec.: J. Transp. Res. Board*, vol. 1944, pp. 82–88, 2006.
[37] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, Mar. 2013.
[38] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recog. Letters*, vol. 27, no. 7, pp. 773–780, 2006.

Zhi Gao received the BEng and PhD degrees from Wuhan University, China, in 2002 and 2007, respectively. Since 2008, he joined the Interactive and Digital Media Institute, National University of Singapore, as a research fellow. His research interests include the topics of structure from motion, change detection, and action recognition.

Loong-Fah Cheong received the BEng degree from the National University of Singapore, and the PhD degree from the University of Maryland at College Park, Center for Automation Research, in 1990 and 1996, respectively. In 1996, he joined the Department of Electrical and Computer Engineering, National University of Singapore, where he is an associate professor currently. His research interests include the processes in the perception of three-dimensional motion, shape, and their relationship, as well as the 3D motion segmentation and the change detection problems.



Yu-Xiang Wang received the BEng degree from the National University of Singapore in 2011. He is currently a research engineer in NUS and at the same time working toward the MEng degree. His research interests include machine learning, computer vision, statistics, optimization, and their applications.