Detection of Moving Object in Dynamic Background Using Gaussian Max-Pooling and Segmentation Constrained RPCA

Yang Li, Guangcan Liu, *Member, IEEE,* Shengyong Chen, *Senior Member, IEEE,*

Abstract—Due to its efficiency and stability, Robust Principal Component Analysis (RPCA) has been emerging as a promising tool for moving object detection. Unfortunately, existing RPCA based methods assume static or quasi-static background, and thereby they may have trouble in coping with the background scenes that exhibit a persistent dynamic behavior. In this work, we shall introduce two techniques to fill in the gap. First, instead of using the raw pixel-value as features that are brittle in the presence of dynamic background, we devise a so-called Gaussian max-pooling operator to estimate a "stable-value" for each pixel. Those stable-values are robust to various background changes and can therefore distinguish effectively the foreground objects from the background. Then, to obtain more accurate results, we further propose a Segmentation Constrained RPCA (SC-RPCA) model, which incorporates the temporal and spatial continuity in images into RPCA. The inference process of SC-RPCA is a group sparsity constrained nuclear norm minimization problem, which is convex and easy to solve. Experimental results on seven videos from the CDCNET 2014 database show the superior performance of the proposed method.

Index Terms—moving object detection, dynamic background, stable-value, low-rank, group sparsity.

I. INTRODUCTION

IN the areas of high safety standards such as stations, airports, schools, banks and so on, surveillance cameras are now ubiquitous, producing a large number of videos every day. The massive nature of surveillance videos makes it very difficult for human investigators to manually search a target through all videos. Therefore, it is urgent to enable the surveillance system to intelligently detect irregularities, suspicious targets, etc. To this end, moving object detection (or background subtraction), which aims to find independent moving objects in a scene, is an important preprocessing step. Many methods have been proposed and investigated in the literature over the past several years, e.g., [1]–[14].

While the background of the scene is fixed and static, there are strong correlations between the video frames. In this case, Robust Principal Component Analysis (RPCA) [15] has already provides us a convenient way to perform moving

Y. Li is with the Department of Computer Science and Engineering, Tianjin University of Technology, 319 Binshuixi Road, Tianjin, China 300384. E-mail: 163102403@stud.tjut.edu.cn.

G. Liu is with B-DAT and CICAEET, School of Information and Control, Nanjing University of Information Science and Technology, 219 Ningliu Road, Nanjing, Jiangsu, China 210044. E-mail: gcliu@nuist.edu.cn. Phone: (86)25-58731276. Fax: (86)25-58731277.

S. Chen is with the Department of Computer Science and Engineering, Tianjin University of Technology, 319 Binshuixi Road, Tianjin, China 300384. E-mail: csy@tjut.edu.cn.
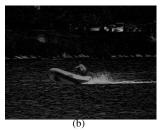

Fig. 1. Exemplifying the performance of RPCA in dealing with dynamic background: (a) an input frame with "surface wave"; (b) detected foreground (white areas). In this example, many background areas are mistaken for foreground moving objects.

object detection. To be more precise, one could stack each frame as a column of a matrix $D \in \mathbb{R}^{m \times n}$ at first, then decompose D into a low-rank term and a sparse term by solving the following convex problem:

$$\min_{A,E}\|A\|_* + \lambda\|E\|_1, \text{ s.t. } D = A + E, \qquad (1)$$

where $\|\cdot\|_*$ is the nuclear norm of a matrix, $\|\cdot\|_1$ denotes the ℓ_1 norm of a matrix seen as a long vector, and $\lambda > 0$ is a parameter. As shown in [15], the low-rank and sparse components correspond to the static background and the moving objects, respectively. However, RPCA relies heavily on the assumption that the background is static or quasi-static and is therefore not applicable to the realistic tasks with dynamic background (see Figure 1).

Some researchers have tried to overcome the drawbacks of RPCA by making use of some additional priors. Zhou et al. [3] considered a prior that the foreground objects are contiguous pieces with relatively small size. This method does improve the identifiability of the foreground objects but still replies on the assumption that the background is quasi-static. Cui et al. [16] extended the RPCA model to process the videos captured by moving cameras. This method works well when the camera motion is predictable, but it cannot handle well the cases where the background scenes exhibit a persistent dynamic behavior, the motion of which is essentially unpredictable. Overall, it is still not enough for existing RPCA based methods to cope with the moving object detection problem in the context of dynamic background.

In this work, we introduce a novel RPCA based method for detecting moving objects in the scenes with dynamic background. As shown in Figure 2, our method contains two key techniques: 1) Instead of using the raw pixels as features

Fig. 2. Outline of the proposed method for moving object detection.

◇ We devise a novel operator called Gaussian max-pooling to estimate the stable-value for each pixel. The stable-values are robust to various background variations such as illumination change, waves in water bodies, falling snow, and so on.

◇ We proposed a novel model termed SC-RPCA, which integrates low-rankness and group sparsity into a unified framework. Different from the existing methods that use the $\ell_{2,1}$ norm to characterize the group sparsity prior [16], [19], [20], SC-RPCA utilizes a generalized $\ell_{2,1}$ to cope with more complicate group sparsity constraints.

◇ Based on the proposed Gaussian max-pooling and SC-RPCA, we establish a novel method for moving object detection in dynamic background. Experimental results show that our method is quite competitive while comparing to the state-of-the-art methods.

The remaining of this paper is organized as follows. Section II summarizes some related works. Section III presents the details of the proposed method. Section IV show empirical results and Section V concludes this paper.

II. RELATED WORK

Due to the correlation between frames, video is an attractive place for low-rank subspaces. As a subspace can be well modeled by a (degenerate) Gaussian distribution [21], it is natural to consider the Gaussian model as a candidate for background modeling. Wren et al. [5] proposed a single Gaussian model to model the background in video frames. Their method procures preferable effects on indoor scenes, but it cannot effectively detect objects in the outdoor which is often a multi-modal environment. Stauffer et al. [6] established an object detection method based on using Gaussian Mixture Model (GMM) [22] to model the background. The processing of each pixel is independent of each other and each pixel is composed of multiple weights of different Gaussian mixture of superposition of distribution. Unlike the single Gaussian model, this method fully uses the historical information to represent the background and therefore can adapt to multi-modal environments. However, its computational cost is too large to promptly deal with sudden background changes. Splitting Gaussian in Mixture Models (SGMM) [7] uses two complementary hybrid Gaussian models with different update rates: One is used to detect moving objects precisely. The other is used to build the background. Comparing to GMM, SGMM can not only improve detection accuracy but also reduce the computational complexity.

Elgammal et al. [8] adopted a Kernel Density Estimation (KDE) for background modeling. The probability density distribution of each pixel is estimated by apply the Gaussian kernel filter to the continuous pixels in all frames. Those densities serve as the basis for judging whether a pixel belongs to a moving object. Generally, this method has strong adaptability and high accuracy. Visual Background Extractor (ViBE) [9] supposes that each pixel has similar distributions as its domain pixels in the spatial domain. The method uses domain pixels to build the background model, which is compared with the currently input pixel values to determine the foreground objects. Pixel-Based Adaptive Segmenter

that are brittle while dealing with dynamic background, we estimate a "stable-value" for each pixel by a novel operator called Gaussian max-pooling. Roughly speaking, Gaussian max-pooling is an elegant technique for selecting the most frequent value from a collection of candidate values. Equipped with those stable-values that sever as good features for distinguishing the foreground objects from the background, RPCA could produce moderately good results by taking as inputs the estimated stable-values rather than the raw pixels. 2) To achieve more accurate results, we shall take into account the temporal and spatial continuity in images. The temporal and spatial continuity are obtained by first oversegmenting the images into superpixels [17], then using the subspace clustering algorithm proposed in [18] to group the superpixels into much larger subregions (image segmentation), and finally using the algorithm in [18] again to group together the similar subregions from different image frames (video segmentation). The segmentation result are considered as a group sparsity constraint and incorporated into RPCA, resulting in a novel model termed Segmentation Constrained RPCA (SC-RPCA). The inference process of SC-RPCA is a group sparsity constrained nuclear norm minimization problem, which is convex and easy to solve. Experimental results on seven videos from the CDCNET 2014 database show the superior performance of the proposed method. In summary, the contributions of this paper include:

TABLE I
EXAMPLES OF THE SAMPLING WINDOWS.

(a) sample 1

46	53	50	51	68
68	68	53	45	49
63	62	**58**	62	48
59	52	43	47	59
82	79	65	62	45

(b) sample 2

46	53	63	62	59
68	69	52	82	79
128	128	**128**	128	128
128	128	128	128	128
128	128	128	128	128

(PBAS) [4] is a non-parametric model based on the gradients of front pixels. It could be robust to slow illumination changes in the background. However, as the spatial continuity is not well considered, this method may fail when dealing with a background of persistent dynamic behavior.

Wang et al. [10] proposed a hybrid, multi-level method termed FTSG, which combines Flux Tensor-based motion detection with the classification results from a Split Gaussian mixture model. Sedky et al. [11] presented a change detection technique based on the dichromatic color reflectance model. Lu et al. [12] used a multi-scale background model for motion detection by following a nonparametric paradigm: Each location in a dynamic scene consisting of a set of samples on different spatial scales. Liang et al. [13] proposed an online object detection method that is robust to sudden illumination changes and regular dynamic background. Recently, Gregorio et al. [14] leveraged Weightless Neural Networks (WNN) to perform moving object detection in dynamic background.

III. OUR METHOD FOR DETECTING MOVING OBJECT IN DYNAMIC BACKGROUND

This section details the proposed techniques, mainly including the approach of Gaussian max-pooling and the model of Segmentation Constrained RPCA (SC-RPCA).

A. Gaussian Max-Pooling

Given a sequence of images $\{I_1, I_2, \cdots, I_N\}$, one could form a data matrix D by stacking each image as a column of D at first, then decompose D into a low-rank term A and a sparse term E by solving the RPCA problem in (1). In the ideal case where the background is strictly static, the term A will be a rank-1 matrix and the nonzero values in E can exactly identify anything that moves. This "strength", however, by itself may become a weakness in the context of dynamic background. As shown in Figure 1, RPCA may wrongly judge many background areas as foreground whenever the background is exhibiting a persistent dynamic behavior.

In order to relieve the drawbacks of RPCA, it is straightforward to exact some features that are robust to background variations. To do this, we would like to consider a simple and effective approach: For a certain pixel v, consider its neighbor pixels in the spatial domain, denoted as $\mathcal{N}_v\{v_1, v_2, \cdots, v_{n^2}\}$, where n is an odd number indicates the window size of sampling. In this work, the sampling window for a pixel is simply a square image region around the pixel, as shown in Table I. Given $\mathcal{N}_v$, the stable-value p_v corresponding to the
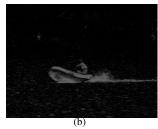

Fig. 3. Exemplifying the effects of Gaussian max-pooling: (a) the input frame shown in Figure 1(a); (b) detection results produced by using the stable-values rather than the pixel-values as inputs for RPCA.

pixel v could be computed as the pixel value that appears most frequently in $\mathcal{N}_v$, i.e.,

$$p_v = \arg \max_{u \in \mathcal{N}_v} \Pr(u).$$

While simple and straightforward, it is indeed not easy to estimate the priori probability of each pixel accurately, as the sampling window is often small (e.g., 5×5). Therefore, we shall propose a Gaussian max-pooling operator to estimate the stable-value of each pixel. Denote by $\mathcal{M}$ all the possible pixel values in image domain, i.e., $\mathcal{M} = \{0, 1, \cdots, 255\}$. For a certain pixel v, we compute its stable-value by maximizing a posterior probability:

$$p_v = \arg \max_{u \in \mathcal{M}} \Pr(\mathcal{N}_v | u) = \arg \max_{u \in \mathcal{M}} \sum_{u' \in \mathcal{N}_v} \Pr(u' | u). \quad (2)$$

As usual, we would assume that the conditional distribution is Gaussian and, accordingly, estimate the conditional probability of a pixel u' given another pixel u by:

$$\Pr(u' | u) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(u' - u)^2}{2\sigma^2}), \quad (3)$$

where $\sigma > 0$ is taken as a parameter.

Algorithm 1 Gaussian Max-Pooling

Input: an image I.
Output: the processed image Ip.
1: Parameters: size of sampling window and σ.
2: **for** each pixel v in I **do**
3:     construct a window $\mathcal{N}_v$ around the pixel v.
4:     compute the conditional probabilities by (3).
5:     compute the stable-value of v by (2).
6: **end for**
7: form a new image Ip by replacing the pixel-values in I with their respective stable-values.
8: **return** Ip

Algorithm 1 summarizes the computational procedures of the Gaussian max-pooling operator. We also give some empirical results in Figure 3. It can be seen that the detection results shown in Figure 3(b) are much better than Figure 1(b). Namely, the background mistaken for foreground in Figure 3(b) is less than Figure 1(b). This demonstrates the effectiveness of our Gaussian max-pooling.
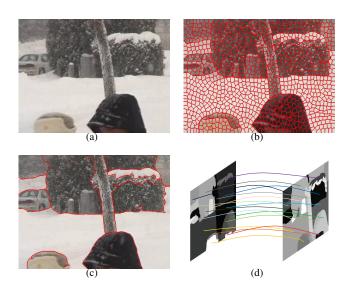
Fig. 4. Obtaining the spatial and temporal continuity in videos: (a) an image; (b) superpixels; (c) results of image segmentation; (d) merging together similar subregions across different frames.

B. Segmentation Constrained RPCA

While better than Figure 1(b), the results in Figure 3(b) are still far from perfect, as there are still many background pixels mistaken for foreground. To obtain more accurate detection, we shall taken into account the spatial and temporal continuity widely existing in videos.

1) Acquiring the spatial and temporal continuity in videos: The desired continuity information is obtained by grouping together the pixels that are similar in appearance and adjacent in the space or time domain. First, we use the approach established in [17] to oversegment the images into superpixels (Figure 4(b)). Then we use the subspace clustering algorithm proposed in [18] to group the superpixels into much larger subregions (Figure 4(c)); this procedure is also known as image segmentation. Finally, we use again the algorithm in [18] to group together the similar, adjacent subregions across different frames (Figure 4(d)). Here, two subregions are considered to be adjacent to each other if and only if: 1) their corresponding frames are adjacent and 2) their regional center distance is smaller than a certain threshold. The whole procedure is called video segmentation and summarized in Algorithm 2.

Algorithm 2 Video Segmentation

Input: a sequence of image frames $\{I_1, \cdots, I_N\}$.
Output: segmentation results
 1: **for** $i = 1, \cdots, N$ **do**
 2:     oversegment I_i into superpixels by [17].
 3:     cluster the superpixles of I_i into subregions by [18].
 4: **end for**
 5: **repeat**
 6:     find all the adjacent subregions across all N frames.
 7:     group together the most similar subregions according to the criterion proposed in [18].
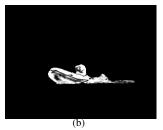 8: **until** no subregions can be grouped

---

Fig. 5. Exemplifying the effects of SC-RPCA: (a) the input frame shown in Figure 1(a); (b) detection results produced by SC-RPCA, using the stable-values computed by Gaussian max-pooling as features.

2) The model of SC-RPCA: Up to now, we have obtained a segmentation for a video (i.e., a sequence of images). For the ease of discussion, we denote the segmentation as

$$C = C_1 \cup C_2 \cup \cdots \cup C_m,$$

where m is the number of groups automatically determined by the algorithm in [18]. For the image pixels belonging to the same group, it is natural to expect that their foregorund/backgournd labels are the same. To this end, we incorporate the constraints encoded by C into RPCA, resulting a novel model termed SC-RPCA:

$$\min_{A,E} \quad \|A\|_* + \lambda \|E\|_{C(2,1)}, \quad \text{s.t. } D_P = A + E, \quad (4)$$

where D_p is D processed by Gaussian max-pooling and $\|\cdot\|_{C(2,1)}$ is the generalized $\ell_{2,1}$ norm associated with a segmentation C. More precisely, the generalized $\ell_{2,1}$ norm is defined by

$$\|E\|_{C(2,1)} = \sum_{i=1}^{m} \sqrt{|C_i| \sum_{(j,k) \in C_i} ([E]_{jk})^2}, \quad (5)$$

where $[\cdot]_{ij}$ is the (i,j)th entry of a matrix, and $|C_i|$ denotes the number of pixels in the group C_i. It is easy to see that, whenever each column in D_p forms a group, $\|\cdot\|_{C(2,1)}$ becomes the traditional $\ell_{2,1}$ norm and therefore SC-RPCA falls back to the RPCA model analyzed in [19].

Thanks to the effects of the low-rankness and group sparsity constraints, SC-RPCA can seamlessly integrate various priors into a unified, convex procedure, including the background correlation, the foreground sparsity, and the temporal/spatial continuity. In additional, the $\ell_{C(2,1)}$ norm is also tolerant to the small mistakes made in the video segmentation procedure. Figure 5 shows an example, which demonstrates the strengths of SC-RPCA.

3) Optimization algorithm: The problem in (4) is convex and can be optimized efficiently with the ALM method [23], which minimizes the following augmented Lagrange function:

$$\mathcal{L}(A, E, Y, \mu) = \|A\|_* + \lambda \|E\|_{C(2,1)} + \langle Y, D_P - A - E \rangle \\ + \frac{\mu}{2} \|D_P - A - E\|_2^F, \quad (6)$$

where Y is Lagrange multipliers and μ is a penalty parameter. The inexact ALM method, which is also called the alternating direction method (ADM), is outlined in Algorithm 3. Notice that the subproblems of the algorithm are convex and have

closed-form solutions. More precisely, Step 3 is solved via the singular value thresholding operator [24], whereas Step 4 is solved as follows:

$$E = \arg\min_E \mathcal{L}(A, E, Y, \mu)$$
$$= \arg\min_E \frac{\lambda}{\mu}\|E\|_{C(2,1)} + \frac{1}{2}\left\|E - \left(D_P - A + \frac{Y}{\mu}\right)\right\|_2^F.$$

Write $M = D_P - A + Y/\mu$, assume that $(j,k) \in C_i$ without loss of generality, and denote $\|[M]_{C_i}\|_2 = \sqrt{\sum_{(a,b)\in C_i}([M]_{ab})^2}$. Then the solution to the above problem is given by

$$[E]_{jk} = \begin{cases} \frac{\|[M]_{C_i}\|_2 - \frac{\lambda|C_i|}{\mu}}{\|[M]_{C_i}\|_2}[M]_{jk}, & \text{if } \|[M]_{C_i}\|_2 > \frac{\lambda|C_i|}{\mu}, \\ 0, & \text{otherwise}, \end{cases}$$

which follows from Lemma 3.2 of [25].

Algorithm 3 Solving Problem (4) by Inexact ALM

Input: feature matrix D_P and segmentation C.
Output: the optimal solution (A, E)
1: initialization: $A = E = Y = 0, \mu = 10^{-6}$.
2: **while** not converged **do**
3:     fix the others and update A by

$$A = \arg\min_A \mathcal{L}(A, E, Y, \mu).$$

4:     fix the others and update E by

$$E = \arg\min_E \mathcal{L}(A, E, Y, \mu).$$

5:     update the multipliers by

$$Y = Y + \mu(D_P - A - E).$$

6:     update the penalty parameter

$$\mu = \rho\mu,$$

    where the parameter ρ takes the role of controlling the convergence speed. It is set as $\rho = 1.1$ in all experiments.
7: **end while**
8: **return** (A, E)

---

4) Overall algorithm for moving object detection: With the optimal solution (A, E) to (4), all the image frames are segmented to foreground and background simultaneously. Namely, the locations of the nonzero elements in E indicate the locations of foreground objects. Algorithm 4 summarizes the whole procedure of our method for detecting moving objects in videos.

IV. EXPERIMENTS AND ANALYSIS

A. Experimental Setting

1) Data: In order to verify the validity of our method, we experiment with seven videos from the CDCNET 2014

Algorithm 4 Moving Object Detection

Input: a video consisting of N images $\{I_1, \cdots, I_N\}$.
Output: the detection results.
1: process all the images by Algorithm 1, resulting in a sequence of processed images $\{Ip_1, \cdots, Ip_N\}$.
2: obtain a video segmentation C by Algorithm 2.
3: form a matrix D_P by stacking each processed image as a column of the matrix.
4: obtain a sparse matrix E by Algorithm 3, using D_P and C as inputs.
5: obtain the final results according to E.

(a)   (b)
(c)   (d)

Fig. 6. Some challenging examples from the CDCNET 2014 database: (a) overpass video. (b) canoe video. (c) snowfall video. (d) winterDriveway video.

database, where the human-annotated ground truth is available. Those videos contain challenging examples in motion detection, e.g., dynamic background (waving trees, surface waves), bad weather (snowfall), intermittent object motion, etc. Figure 6 shows some challenging examples. Note that there are waving trees in overpass video that cause the background to change. In Figure 6(c), the snowfall affects the integrity of moving object detection. Regarding the example in Figure 6(d), the intermittent object motion makes the pixels of moving object unchanged in several consecutive frames .

2) Evaluation metrics: The following three measures are used for evaluation: 1) True Positive (TP), which denotes the number of true foreground pixels correctly classified as foreground. 2) True Negative (TN), which is the number of foreground pixles wrongly classified as background. 3) False Positive (FP), which is the number of background pixels wrongly classified as foreground. Then, Recall, Precision and F-measure are used to evaluate various moving object detection algorithms:

$$Recall = \frac{TP}{TP + TN} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$F - measure = 2\frac{Recall * Precision}{Recall + Precision} \quad (9)$$

3) Baselines: To verify the effectiveness of the proposed Gaussian max-pooling operator and the SC-RPCA model, we include for comparison the RPCA based methods, including "RPCA+pixels" (which uses raw pixels as inputs for RPCA) and "RPCA+stable-value" (which takes the state-values estimated by our Gaussian max-pooling as inputs). To show the superiority of our moving object detection method, we also include for comparison several state-of-the-art methods, including XUE [26], Euclidean distance (ED) [27], MST [12], Spectral [11], FSTG [10], CwisarDH [14], GMM [22], CP3online [13], BinWang [28] and KDE [8].

B. Results

1) Comparison with RPCA based methods: Table II shows the evaluation results on the snowfall video. It is can be seen that the original approach of RPCA+pixel is largely outperformed by RPCA+stable-value, which is further distinctly outperformed by SC-RPCA+stable-value. These results illustrate that both the Gaussian max-pooling operator and the SC-RPCA model are effective in detecting moving objects in dynamic background.

TABLE II
COMPARISON RESULTS ON THE SNOWFALL VIDEO.

Measure	RPCA+pixel	RPCA+stable-value	SC-RPCA+stable-value
Recall	0.128	0.361	0.957
Precision	0.949	0.531	0.787
F-measure	0.226	0.430	0.863

2) Comparison with state-of-the-art methods: Table III shows evaluation results on seven videos from the CDCNET 2014 database. It can be seen that, except for the overpass and winterDriveway videos, the proposed method achieves a F-measure that is similar to or higher than the competing methods.

Figure 7 gives some examples of the detection results, demonstrating the ability of the proposed method in dealing with complex dynamic scenes. Regarding the first column that shows car moving in the snowy weather, only CP3˙online [13] and the proposed method perform well, while Xue [26] and MST [12] lost many foregrounds. In the second column that also shows car moving in the snowy weather (but the weather is better than the first column), Spectral [11], FTSG [10], CwisarDH [14], GMM [22], CP3˙online [13] and the proposed method have good results. While dealing with the third column that shows people skating in the snowy weather, Xue [26] loses many foregrounds in the scenes "Skating", "snowfall" and "Blizzard". On the forth and fifth columns that show boats and canoe in dynamic water surface, Xue [26] loses many foregrounds in the scenes "canoe" and "boats", while Euclideandistance [27], GMM [22], CP3˙online [13] and KDE [8] wrongly detect some background as foreground. The sixth column shows a man walking in the scene with swaying leaves, where BinWang [28] gets better results. In the seventh column with intermittent motion, GMM [22] loses much foreground.
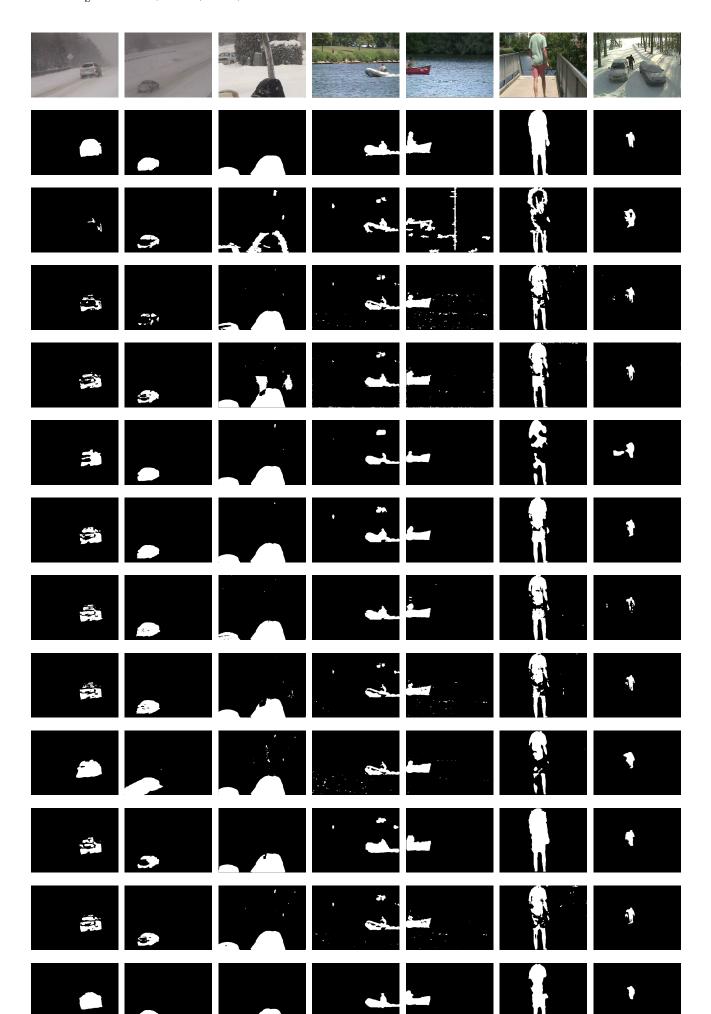
V. CONCLUSION

RPCA is sensitive to dynamic background and bad weather. To overcome this problem, firstly, instead of using the raw pixel-value as features that are brittle while applying to dynamic background, a Gaussian max-pooling operator is used to estimate the stable-value for each pixel. Those stable-pixels are robust to various background. Then, the video segment is used to incorporate the temporal and spatial continuity of both the foreground and background for group sparsity constrain. The proposed method is tested in some challenging situations. Compared with other popular methods, the experimental results demonstrate the proposed has better performance for the dynamic background and bad weather.

REFERENCES

[1] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector cnns," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 2718–2726.

[2] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele, "Monocular visual scene understanding: Understanding multi-object traffic scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 882–897, April 2013.

[3] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 597–610, March 2013.

[4] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 2012, pp. 38–43.

[5] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, Jul 1997.

[6] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 1999, pp. 252–259.

[7] R. H. Evangelio, M. Pätzold, and T. Sikora, "Splitting gaussians in mixture models," in *IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, Sept 2012, pp. 300–305.

[8] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," vol. 1843, 2000.

[9] O. Barnich and M. V. Droogenbroeck, "Vibe: A powerful random technique to estimate the background in video sequences," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 945–948.

[10] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Static and moving object detection using flux tensor with split gaussian models," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2014, pp. 420–424.

[11] M. Sedky, M. Moniri, and C. C. Chibelushi, "Spectral-360: A physics-based technique for change detection," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2014, pp. 405–408.

[12] X. Lu, "A multiscale spatio-temporal background model for motion detection," in *2014 IEEE International Conference on Image Processing*, Oct 2014, pp. 3268–3271.

[13] D. Liang and S. Kaneko, "Improvements and experiments of a compact statistical background model," *ArXiv e-prints*, May 2014.

[14] M. D. Gregorio and M. Giordano, "Change detection with weightless neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2014, pp. 409–413.

TABLE III
EVALUATION RESULTS ON SEVEN VIDEOS FROM THE CDCNET 2014 DATABASE.

Video	Measure	[26]	[27]	[12]	[11]	[10]	[14]	[22]	[13]	[28]	[8]	proposed
snowfall	Recall	0.360	0.993	0.990	0.637	0.699	0.859	0.630	0.640	0.986	0.986	0.955
	Precision	0.802	0.358	0.496	0.976	0.982	0.844	0.955	0.978	0.592	0.613	0.930
	F-measure	0.497	0.529	0.665	0.758	0.812	0.843	0.751	0.763	0.738	0.754	**0.942**
blizzard	Recall	0.693	1.000	0.510	0.753	0.943	0.997	0.915	0.972	0.998	0.995	0.933
	Precision	0.875	0.247	0.999	0.962	0.969	0.491	0.930	0.901	0.618	0.686	0.979
	F-measure	0.769	0.385	0.661	0.817	0.954	0.652	0.922	0.935	0.756	0.806	**0.955**
skating	Recall	0.562	1.000	0.670	0.938	0.991	0.993	0.984	0.962	0.997	0.985	0.995
	Precision	0.426	0.558	0.948	0.913	0.960	0.978	0.951	0.966	0.968	0.972	0.947
	F-measure	0.478	0.681	0.778	0.925	0.975	**0.986**	0.967	0.964	0.982	0.979	0.970
Boats	Recall	0.845	0.811	0.806	0.702	0.830	0.875	0.543	0.730	0.730	0.682	0.957
	Precision	0.793	0.718	0.922	0.991	0.941	0.865	0.930	0.757	0.919	0.957	0.787
	F-measure	0.818	0.761	0.859	0.820	**0.880**	0.864	0.685	0.730	0.813	0.796	0.863
canoe	Recall	0.516	0.840	0.708	0.626	0.838	0.749	0.638	0.767	0.957	0.672	0.865
	Precision	0.308	0.763	0.937	0.994	0.966	0.989	0.994	0.956	0.859	0.951	0.944
	F-measure	0.387	0.792	0.805	0.767	0.896	0.852	0.776	0.851	**0.905**	0.786	0.901
overpass	Recall	0.424	0.988	0.700	0.555	0.842	0.690	0.748	0.571	0.952	0.656	0.886
	Precision	0.819	0.624	0.992	0.980	0.982	0.993	0.769	0.996	0.917	0.980	0.830
	F-measure	0.559	0.764	0.820	0.704	0.906	0.815	0.759	0.723	**0.934**	0.786	0.851
winterDriveway	Recall	0.481	1.000	0.951	0.572	0.895	0.917	0.931	0.729	0.839	0.877	0.734
	Precision	0.917	0.561	0.755	0.998	0.932	0.593	0.794	0.926	0.881	0.849	0.993
	F-measure	0.630	0.714	0.840	0.727	**0.913**	0.711	0.857	0.814	0.858	0.862	0.843

[15] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[16] X. Cui, J. Huang, S. Zhang, and D. N. Metaxas, "Background subtraction using low rank and group sparsity constraints," in *European conference on Computer Vision*, vol. 1, October 2012, pp. 612–625.

[17] G. Mori, "Guiding model search using segmentation," in *IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1417–1423.

[18] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1546–1562, 2007.

[19] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3047–3064, 2012.

[20] G. Liu, H. Xu, J. Tang, Q. Liu, and S. Yan, "A deterministic analysis for LRR," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 38, no. 3, pp. 417–430, 2016.

[21] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[22] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *International Conference on Pattern Recognition*, vol. 2, Aug 2004, pp. 28–31.

[23] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," UIUC Technical Report UILU-ENG-09-2215, Tech. Rep., 2009.

[24] J. Cai, E. Candes, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[25] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *International Conference on Machine Learning*, 2010, pp. 663–670.

[26] Y. Xue, X. Guo, and X. Cao, "Motion saliency detection using low-rank and sparse decomposition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2012, pp. 1485–1488.

[27] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Comparative study of background subtraction algorithms," *Journal of Electronic Imaging*, vol. 19, no. 3, p. 033003, 2010.

[28] B. Wang and P. Dudek, "A fast self-tuning background subtraction algorithm," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2014, pp. 401–404.

Yang Li received the bachelor's degree from Nanjing University of Information Science and Technology, Nanjing, China, in 2013, and the master's degree from Beifang Univesity of Nationalities, Yinchuan, China, in 2016. Now he is a PhD candidate in Tianjin University of Technology. His research interests touch on the areas of computer vision, and image processing.

Guangcan Liu (M'11) received the bachelor's degree in mathematics and the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2004 and 2010, respectively. He was a Post-Doctoral Researcher with the National University of Singapore, Singapore, from 2011 to 2012, the University of Illinois at Urbana-Champaign, Champaign, IL, USA, from 2012 to 2013, Cornell University, Ithaca, NY, USA, from 2013 to 2014, and Rutgers University, Piscataway, NJ, USA, in 2014. Since 2014, he has been a Professor with the School of Information and Control, Nanjing University of Information Science and Technology, Nanjing, China. His research interests touch on the areas of machine learning, computer vision, and image processing.

Shengyong Chen (M'01 - SM'10) Shengyong Chen received the Ph.D. degree in computer vision from City University of Hong Kong, Hong Kong, in 2003. He is currently a Professor of Tianjin University of Technology and Zhejiang University of Technology, China. He received a fellowship from the Alexander von Humboldt Foundation of Germany and worked at University of Hamburg in 2006 - 2007. His research interests include computer vision, robotics, and image analysis. Dr. Chen is a Fellow of IET and senior member of IEEE and CCF. He has published over 100 scientific papers in international journals. He received the National Outstanding Youth Foundation Award of China in 2013.