

Research Article

Compressive Background Modeling for Foreground Extraction

Yong Wang,¹ Qian Lu,² Dianhong Wang,^{1,2} and Wei Liu¹

¹Faculty of Mechanical and Electronic Information, China University of Geosciences, Wuhan 430074, China

²Department of Geological Science and Engineering, Wuhan University of Engineering Sciences, Wuhan 430200, China

Correspondence should be addressed to Yong Wang; wycug2011@gmail.com

Received 13 October 2014; Revised 10 February 2015; Accepted 11 March 2015

Academic Editor: Jiri Jan

Copyright © 2015 Yong Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Robust and efficient foreground extraction is a crucial topic in many computer vision applications. In this paper, we propose an accurate and computationally efficient background subtraction method. The key idea is to reduce the data dimensionality of image frame based on compressive sensing and in the meanwhile apply sparse representation to build the current background by a set of preceding background images. According to greedy iterative optimization, the background image and background subtracted image can be recovered by using a few compressive measurements. The proposed method is validated through multiple challenging video sequences. Experimental results demonstrate the fact that the performance of our approach is comparable to those of existing classical background subtraction techniques.

1. Introduction

Foreground extraction is often the first step of many visual surveillance applications such as object tracking, recognition, and anomaly detection. Background subtraction [1] is the most frequently used method to detect and extract objects automatically in video sequences. The basic principle can be formulated as a technique that builds the background model and compares this model with the current image frame in order to distinguish foreground, that is, moving objects from static or slow moving background. Many pixel-based methods have been investigated in the past decades. Among these, Gaussian mixture model (GMM) [2] is a representative method for robustly modeling complicated backgrounds with slow illumination changes and small repetitive movements. This method models the distribution of the values observed over time at each pixel by a weighted mixture of Gaussians. Several modified methods concerning the number of Gaussian components, learning rate, and parameters update [3–5] have been proposed. However, the main drawback of these approaches is being computationally intensive. In [6], a nonparametric kernel density estimation (KDE) has been proposed to model the background, but this method consumes too much memory. Besides, in [7], each pixel is represented by a codebook which is able to

capture background motion over a long period of time with a limited amount of memory. The codebooks can evolve with illumination variations and moving backgrounds. Nevertheless, the codebook update will not allow the creation of new codewords once codebooks have been learned from a typically training sequence. Recently, a universal background subtraction method called ViBe has been presented in [8], where each pixel is modeled with a set of real observed pixel values. It is reported that this method outperforms current mainstream methods in terms of both computation speed and detection accuracy.

Most of pixel-level background subtraction techniques have considerable computation costs; thus some other methods are proposed to improve the computation efficiency. Our earlier work [9] proposed a block-based image reconstruction to reduce the video frame size and then applied GMM to model the background for the constructed frame. In recent years, there has been a growing interest in compressive sensing (CS) and the idea of CS has also been exploited for background subtraction. In [10], an image is divided into small blocks and random projections based on CS are then computed for each block to reduce the data dimensionality. After this, each projection value is modeled by GMM to determine whether the block belongs to foreground or not. The work in [11] describes a CS reconstruction

method to directly recover background subtracted images from compressive measurements. The main idea is that the background subtracted images can be represented sparsely in the spatial image domain. Based on this, an improved method called adaptive rate compressive sensing (ARCS) is presented in [12]. Furthermore, a dynamic group sparsity (DGS) recovery algorithm [13] is proposed based on the extended CS theory to perform background subtraction. According to both sparsity and dynamic group clustering priors, DGS can recover stably background subtracted image using fewer measurements and computations.

Recently, the approach proposed in [14] exploits sparse representation (SR) to perform classification, where a test image can be sparsely represented by a subspace spanned from training images. CS has shown the ability to efficiently compress signals using SR. Motivated by this idea, we present a new background subtraction method combining CS and SR techniques in this paper. In our solution, the background modeling can be cast as a greedy sparse recovery problem. This paper is organized as follows. In Section 2, we briefly review the CS theory. This review presents the framework developed for compressive background modeling. Section 3 extensively describes our solution and the implementation process: sparse representation, background initialization, background modeling, and update mechanism. Section 4 compares our method with several state-of-the-art background subtraction techniques and discusses the performances in terms of accuracy and computation efficiency. Section 5 concludes this paper.

2. Compressive Sensing

The CS theory [15] states that a signal can be reconstructed from a small number of measurements with high probability, provided that the signal is sparse in the spatial domain or some transform domains, for example, wavelets. Assume that a signal x can be represented as $x = \Psi\theta$, where Ψ denotes a basis and θ is the coefficients corresponding to the basis. The signal x is said to be k -sparse if all other elements in θ vanish except for k nonzero coefficients. According to CS, for a sparse signal $x \in R^n$, compressive measurements can be collected by the following random projections:

$$y = \Phi x + \varepsilon, \quad (1)$$

where $\Phi \in R^{m \times n}$, $m \ll n$, is the measurement matrix, $y \in R^m$ contains m measurements, and ε is the measurement noise. Specifically, a high dimensional vector x is converted into a much lower dimensional measurement vector y . Moreover, the compressive measurements in y contain almost all the information of the sparse vector x . This means that we can work with data of significantly lower dimension so as to achieve computation efficiency as well as accuracy.

Since $m \ll n$, the recovery of the sparse signal x from y is underdetermined. However, the following two additional assumptions make the recovery possible [16, 17]. First, Φ is required to satisfy the restricted isometry property (RIP). Some randomly generated matrices, for example, Gaussian, Bernoulli distribution of ± 1 , obey the RIP and can be used for

random projections. Second, m is greater than $O(k \log(n/k))$; that is, the number of measurements must be large enough with respect to the number of nonzero coefficients in θ . Under these circumstances, the sparse recovery can be formulated by solving the following ℓ_0 minimization problem:

$$\hat{x} = \operatorname{argmin} \|x\|_0 \quad \text{s.t.} \quad \|y - \Phi x\|^2 < \varepsilon. \quad (2)$$

It is known that (2) is an NP hard problem. In general, the sparse solution can be obtained by ℓ_1 convex optimization or greedy iterative algorithms.

3. Compressive Background Modeling

3.1. Sparse Representation of Background. To improve the computation efficiency of sparse recovery, the image frame is first divided into M blocks of pixels. We describe the sparse representation of background on a block because the operations to be carried out on the vector of each block are identical. For the i th block, $i = 1, \dots, M$, it is vectorized into a column vector x_i of size $n \times 1$, where n denotes the product of the height and width of the block. The vectorized image block x_i is assumed to consist of both background b_i and background subtracted image f_i ; that is,

$$x_i = b_i + f_i. \quad (3)$$

According to SR, the current background can be represented as a linear combination of all preceding background images. Due to the memory requirement blast for long video sequences, we use l preceding backgrounds to sparsely represent the new background, where l should be large enough to comprise a sufficient number of background samples. Thus the background b_i can be measured as follows:

$$b_i = A_i \alpha_i = [b_{i,1}, b_{i,2}, \dots, b_{i,l}] \alpha_i, \quad (4)$$

where $A_i \in R^{n \times l}$ is a subspace spanned from preceding backgrounds $b_{i,j}$, $j = 1, \dots, l$. $\alpha_i \in R^l$ is a sparse coefficient vector whose most elements are either zeros or very close to zeros. Substituting (4) into (3), we have

$$x_i = A_i \alpha_i + f_i = [A_i, I] \begin{bmatrix} \alpha_i \\ f_i \end{bmatrix} = \Psi_i s_i, \quad (5)$$

where $I \in R^{n \times n}$ is an identity matrix, $\Psi_i \in R^{n \times (l+n)}$ is called the dictionary, and $s_i \in R^{l+n}$ is the corresponding coefficient vector. In fact, the sizes of the foreground objects are relatively smaller than the size of the background image. Only a small part of pixels in the background subtracted image has nonzero values; that is, f_i exhibits sparsity in the spatial domain. The fact that both α_i and f_i are sparse implies that s_i is sparse. In this case, x_i can be well approximated using nonzero coefficients of s_i under Ψ_i , which is critical to the sparse recovery base on CS.

3.2. Background Initialization. As described before, the background subspace is characterized by A_i containing l preceding backgrounds. Thus the first problem we face

is the initialization of A_i . Normally, most mainstream approaches need tens or hundreds of image frames to initialize background by estimating probability density functions or statistical parameters of background pixels. For instance, GMM initializes every background pixel by a mixture of K Gaussian distributions, where the weight, mean, and variance of each distribution are learned and updated from a sequence of frames. Nevertheless, the model initialization takes up large amounts of computation resources. Besides, the shape of a probability density function is sensitive to outliers and the evaluation of the statistical parameters is largely dependent upon the number of samples to be considered. As a matter of fact, it is not imperative to estimate the temporal distribution of background pixels using a large amount of pixel samples. Each background pixel can be initialized with a set of real observed pixel values instead of with an explicit pixel model.

Like the authors of [8], we initialize A_i from the first frame of the video sequence based on the assumption that neighboring pixels share a similar temporal distribution. Given a pixel located at u in i th image block, its value and the spatial neighborhood are denoted by $v(u)$ and $N_G(u)$, respectively. Therefore, the background sample value of the pixel u , $b_{i,j}(u)$, is equal to the value of a pixel v randomly selected in $N_G(u)$ in the first frame:

$$b_{i,j}(u) = v(v | v \in N_G(u)) \quad j = 1, \dots, l. \quad (6)$$

Based on this, the background model of pixel u is filled with a collection of l sample values. Thus A_i can be initialized by background models of all pixels in the i th block:

$$A_i = [\{b_{i,1}(u)\}, \{b_{i,2}(u)\}, \dots, \{b_{i,l}(u)\}], \quad u \in \text{block } i. \quad (7)$$

It is noteworthy that this strategy can perform foreground extraction from the second frame, which is beneficial for a short video sequence or memory-constrained embedded devices. Furthermore, the spatial neighborhood of moderate size is preferred. A large size leads to the degradation in the statistical correlation between pixels at different locations. Conversely, the diversity of background samples cannot be guaranteed under a small size. In our solution, samples are selected randomly in the 8-connected neighborhood of each pixel.

3.3. Compressive Background Modeling. Likewise, we take the i th image block for example. As mentioned earlier, due to the sparsity of s_i under the dictionary Ψ_i , x_i can be projected into compressive measurements without losing much information. Combining (5) with (1), random projections are executed for each block:

$$y_i = \Phi x_i + \varepsilon = \Phi [A_i, I] \begin{bmatrix} \alpha_i \\ f_i \end{bmatrix} + \varepsilon = \Phi \Psi_i s_i + \varepsilon. \quad (8)$$

Let Φ be an $m \times n$ random matrix whose entries $\varphi_{p,q}$ are independent realizations of ± 1 Bernoulli random variables:

$$\varphi_{p,q} = \begin{cases} +\frac{1}{\sqrt{m}}, & \text{with probability } \frac{1}{2} \\ -\frac{1}{\sqrt{m}}, & \text{with probability } \frac{1}{2}. \end{cases} \quad (9)$$

Note that once Φ is generated in the beginning, the same matrix is used for each block throughout the video sequence.

Assume that $\Psi_{t,i}$ is the dictionary of the i th block at time t . For the image at time $t + 1$, let $x_{t+1,i}$ be the vector of the i th block. Given Φ described in (9), the compressive measurements $y_{t+1,i}$ can be obtained by the matrix multiplication of Φ and $x_{t+1,i}$. According to (8), the background modeling at time $t + 1$ is thus formulated as the following sparse recovery problem:

$$(\alpha_{t+1,i}, f_{t+1,i}) = \operatorname{argmin} \|s_{t,i}\|_0, \quad \text{s.t. } \|y_{t+1,i} - \Phi \Psi_{t,i} s_{t,i}\|^2 < \varepsilon. \quad (10)$$

In our method, the sparse recovery in (10) is solved by a greedy algorithm called DGS [13]. The reason for this is that DGS can recover stably sparse data with less measurement requirement and lower computation complexity, because it considers not only sparsity but group clustering priors of sparse data as well. Compared to existing greedy recovery algorithms, the difference is that the estimation pruning uses DGS approximation rather than k -sparse approximation. To reduce the computational cost, background modeling can be performed at specified intervals instead of every frame.

3.4. Background Update. To adapt to gradual changes that occur in the background such as environment disturbances and slow lighting changes, the background model needs to be updated over time. In this paper, we tackle this problem by dynamically updating background model once every T frames, and the frame interval is set at 3 to 7. The reason for this is threefold. Firstly, it is acceptable that background model remains the same for a certain period of time, which does not materially deteriorate the results of background subtraction. Secondly, the continuous sparse recovery results in a relatively heavy computation overhead. Furthermore, too often update introduces small errors into background model at each time, and the accumulative errors have negative effects on the accuracy of foreground extraction.

At initialization, A_i is built using (7) from the first video frame. Subsequently, A_i is updated according to the sparse recovery result. Assume that the background subspace at time t is denoted as $A_{t,i}$. The sparse coefficient vector at time $t + 1$, $\alpha_{t+1,i}$, can be obtained via (10); thus the background at time $t + 1$ is calculated by $b_{t+1,i} = A_{t,i} \alpha_{t+1,i}$. We can see from (4) that the background update is equivalent to replacing a certain sample $b_{i,j}$, $j = 1, \dots, l$, in $A_{t,i}$ using $b_{t+1,i}$. The traditional approach to background update is to substitute the new values for old ones in turn. However, there is no reason to naively remove a valid sample if it corresponds to a background. Unlike this idea, we adopt a probability strategy to update $A_{t,i}$ where a sample in $A_{t,i}$ is chosen at random according to a uniform law and replaced by the new background $b_{t+1,i}$. Therefore the background subspace at time $t + 1$ is denoted by

$$A_{t+1,i} = A_{t,i} \left\{ \forall b_{i,j} \mid b_{i,j} = b_{t+1,i}, \text{ with probability } \frac{1}{l} \right\}. \quad (11)$$

Furthermore, the background update should also take sudden illumination changes into account. As stated earlier,

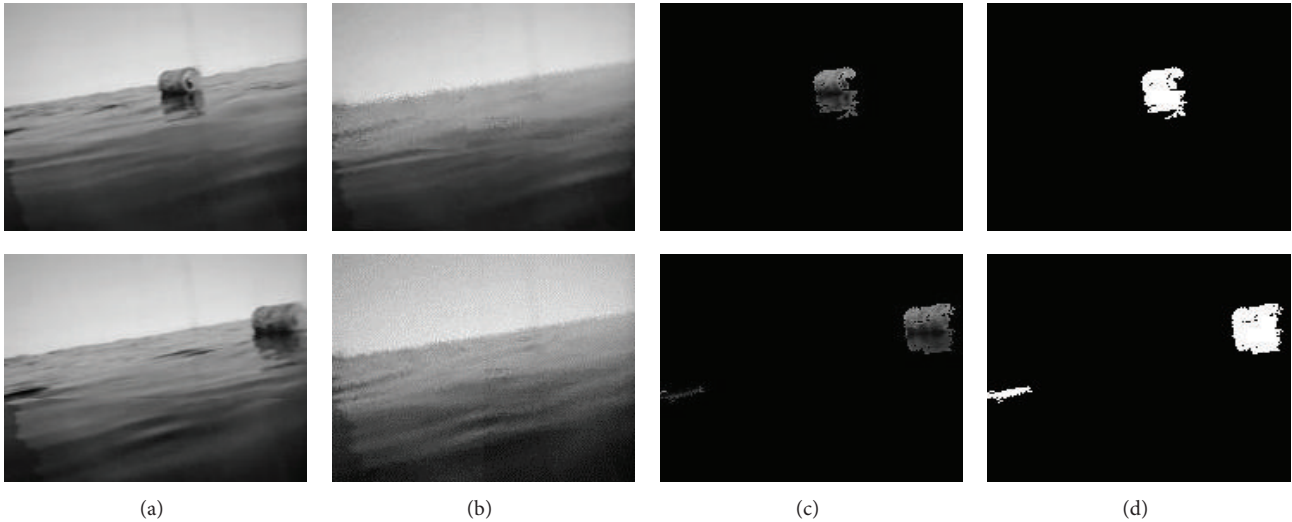


FIGURE 1: Results on a video sequence of 150 frames with background disturbances. (a) Original frames, (b) background, (c) background subtracted images, and (d) foreground masks.

A_i is initialized from a single frame. This technique provides a rapid response to sudden illumination changes; that is, the existing background model is discarded directly and a new model is initialized right now. Obviously, a sudden illumination change leads to the great change of pixel value. We count the pixels with great value change; if the ratio of these pixels to the total exceeds a threshold T_h , the background is reinitialized according to the current frame in order to accomplish foreground extraction under the new illumination condition.

4. Experimental Results

4.1. Effectiveness Analysis. This experiment is designed to demonstrate the effectiveness of compressive background modeling. We implement the proposed method in MATLAB and test it on two complicated scenarios involving background disturbances and illumination changes. The same parameters are used for these sequences in the experiment. The values of parameters are as follows: $M = 4$, $l = 15$, $T = 5$, $T_h = 0.6$, and the measurement rate $m/n = 1/16$.

Figure 1 shows examples of background modeling and foreground extraction for two typical frames of the first sequence of 176×144 pixels, which concerns a floating tin can under condition of waves. Figure 1(a) is the original frames, Figures 1(b) and 1(c) show the recovered background images and background subtracted images, respectively. To facilitate the observations, Figure 1(d) gives the binary foreground masks that are directly obtained via background subtracted images. All results are the solutions of the optimization problem in (10) with a simple postprocessing based on morphological operations. As can be seen, the proposed method can almost accurately recover the background and extract foreground objects, except in the case of severe disturbances; for example, the background subtracted image shown in the second row contains a small piece of background corresponding to a heavy wave. The results show that the

background modeling based on CS and SR techniques can handle dynamic scenes well. Moreover, the spatial consistency of neighboring pixels is absorbed into the initialization of background model, which can mitigate the impact of waves to some extent.

Figure 2 shows the background subtraction results on the other video sequence of 160×120 pixels, which concerns three different illumination conditions, that is, normal; the light is switched off and switched on again. As described in Section 3.4, when the light is switched off/on, a new background model is initialized instantaneously to keep track of the sudden change in illumination. After this, the recovery of the background and background subtracted images can proceed as before and continue to improve. From Figure 2, it is clear that the resulting images are clean even undergoing repetitive sudden illumination changes. This means that the proposed method is able to handle random illumination conditions. In addition, our method overcomes effectively the impact of the flickering monitors.

4.2. Qualitative Comparison. To evaluate the performances of the proposed method, we compare the results of our method with those of five representative background subtraction algorithms: (1) ARCS [12]; (2) CS-MoG [10]; (3) ViBe [8]; (4) KDE [6]; and (5) GMM [2]. We implement the five algorithms by ourselves, and all the parameters in these algorithms use the proposed default values according to the original papers. These algorithms are tested on seven different video sequences, and all the experiments give similar conclusions as described below. Due to space constraints, only five out of seven sequences are illustrated in Figure 3. The first three sequences involving pedestrians and vehicles detection under sunny and overcast days contain 500 frames of 320×240 pixels. The last two sequences concern the foreground extraction in the presence of background disturbances such as swaying tree and waves on the water, where the former is a standard sequence consisting of 286



FIGURE 2: Results on a video sequence of 750 frames with sudden illumination changes. (a) Original frames, (b) background, (c) background subtracted images, and (d) foreground masks.

frames of 160×120 pixels and the latter comprises 150 frames of 176×144 pixels. According to the image size, the number of image blocks is set to $M = 16$ for the first three sequences and $M = 4$ for the last two sequences. Figure 3(a) shows one typical frame of each sequence, and Figures 3(b)–3(g) correspond to the foreground extraction results of the aforementioned six methods. Note that all the foreground masks are immediate results without any morphological postprocessing operations. Thus the comparisons based on these results are objective and reasonable.

There are several points we can learn from the experimental results. First, the results of our method and ARCS are similar due to the fact that background subtraction is cast as a sparse recovery problem. Both of them have relatively stable performance under various scenarios, as shown in Figures 3(b) and 3(c). Second, CS-MoG is a block-based background subtraction algorithm. The image is segmented into blocks of 8×8 pixels, and projections based on CS are computed for each block. Each projection value is modeled as a GMM to determine if the block is classified as foreground. This makes CS-MoG able to effectively eliminate background disturbances and correctly identify foreground regions. Meanwhile, it also leads to some problems. For example, the foreground masks are not accurate enough because of many holes and fragments. Besides, we can see from the third foreground mask in Figure 3(d) that vehicles with small size are omitted because of the block-based processing. Third, for the first two sequences, the results of ViBe look better. However, the foreground extraction of ViBe evidently deteriorates when there exist severe background disturbances. Fourth, KED and GMM are more sensitive to illumination changes and cannot deal with nonstatic scenes. From Figures 3(f) and 3(g), there

are a large number of background pixels incorrectly classified as foreground under the condition of swaying tree and waves. Comparatively, our method is stable and effective under different illumination conditions and cluttered backgrounds.

Two metrics, the percentage of correct classification (PCC) [8] and the average processing time (APT), are utilized to quantitatively evaluate the six background subtraction algorithms. Naturally, the higher the PCC, the more accurate the foreground extraction; in the meanwhile, the less the APT, the higher the computation efficiency. The comparison results are computed as the mean of 50 consecutive frames. Figures 4 and 5 illustrate the PCCs and APTs of the six methods for five sequences, respectively. We can see from Figure 4 that the PCCs confirm the results illustrated in Figure 3. The PCC of ARCS is very close to that of our method under different scenarios. In most cases, our method has the best performance and its PCC is higher than those of the others. For one thing the PCC of ViBe is slightly greater than that of our method for the first two sequences and for another the PCC of ViBe reduces by an average of 7.48% in comparison with our method for the rest sequences. Moreover, the PCC of our method is 4.51 to 10.04% greater than that of CS-MoG.

Though our method and ARCS perform foreground extraction by solving sparse recovery that is relatively time-consuming, our method is faster than ARCS, as shown in Figure 5. The reason for this is twofold. First, in our solution, the image frame is processed in the form of subblocks of pixels instead of a whole image so that the computation efficiency of sparse recovery is improved. Second, we solve the sparse recovery using DGS with lower computation complexity and fewer measurements. As can be seen from Figure 5, CS-MoG which uses random projections to reduce



FIGURE 3: Comparative foreground masks of six methods. (a) Original frames, (b) our method, (c) ARCS, (d) CS-MoG, (e) ViBe, (f) KDE, and (g) GMM.

data dimensionality is significantly faster than other pixel-level techniques, that is, ViBe, KDE, and GMM. For KDE and GMM, background models maintain complete information on each pixel, which results in a considerable computation cost. For instance, for the first sequence, the APTs of our method, ARCS, CS-MoG, ViBe, KDE, and GMM, are 90.91, 129.87, 52.37, 79.07, 154.46, and 122.78 ms, respectively.

Specifically, our method is approximately 42.7%, 69.7%, and 34.9% faster than ARCS, KDE, and GMM, respectively. On the other side, our method obtains foreground by solving optimization problem but CS-MoG and ViBe do not; therefore our method is 73.4% and 14.9% slower than CS-MoG and ViBe, respectively. Furthermore, we further observe the time consumption of our method. The test shows that the sparse

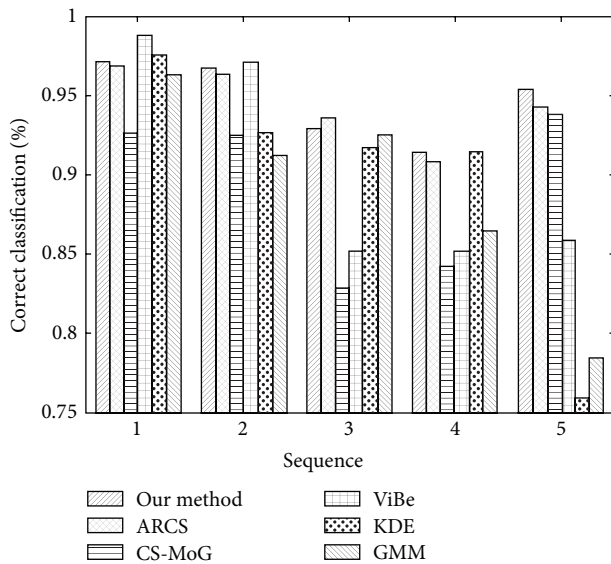


FIGURE 4: PCCs of six methods.

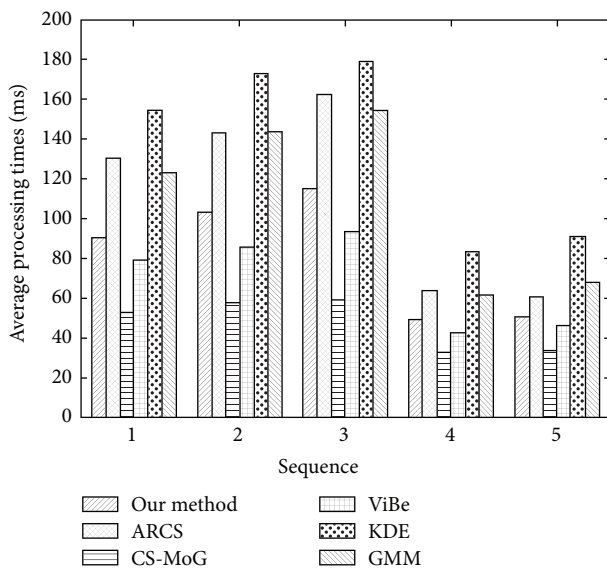


FIGURE 5: APTs of six methods.

recovery dominates the computation time and consumes as much as approximately 85.2% of total computation time. In comparison, the overhead introduced by background update is 10.6% of total computation overhead.

From the foregoing, our method is slower than CS-MoG and ViBe, while the foreground extraction accuracy of our method is superior to those of two others. It can be concluded that the proposed method achieves the best tradeoff between the accuracy and computation efficiency.

5. Conclusions

In this paper, we propose a background subtraction method based on CS and SR. Background is modeled as a sparse representation of preceding backgrounds. Combining the sparse representation of background with the sparsity of

foreground, we use a few compressive measurements to recover background and the background subtracted image within the CS framework. Moreover, we provide the background initialization and update scheme which improve the robustness against the changes in the scenarios. According to comparisons and analyses involving several challenging sequences and five other state-of-the-art background subtraction algorithms, the experimental results clearly demonstrate the effectiveness of the proposed method.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

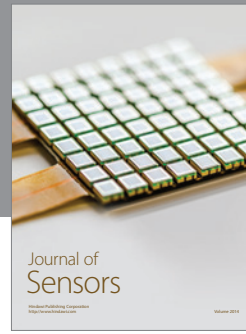
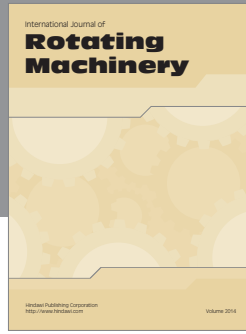
This work was supported in part by Technology Research Program of Hubei Province, China (Grants 2012FFA108 and 2013BHE009), Wuhan Youth Chenguang Program of Science and Technology (Grant 2014070404010209), and Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan).

References

- [1] Y. Benezeth, P. M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, pp. 1–4, Tampa, Fla, USA, December 2008.
- [2] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, pp. 246–252, Fort Collins, Colo, USA, June 1999.
- [3] Y. Wang, J. W. Tian, and Y. H. Tan, "Effective Gaussian mixture learning and shadow suppression for video foreground segmentation," in *MIPPR 2007: Automatic Target Recognition and Image Analysis; and Multispectral Image Acquisition, 67861D*, vol. 6786 of *Proceedings of SPIE*, Wuhan, China, November 2007.
- [4] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 2, pp. 28–31, Washington, DC, USA, 2004.
- [5] J. Cheng, J. Yang, Y. Zhou, and Y. Cui, "Flexible background mixture models for foreground segmentation," *Image and Vision Computing*, vol. 24, no. 5, pp. 473–482, 2006.
- [6] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Computer Vision—ECCV 2000: 6th European Conference on Computer Vision Dublin, Ireland, June 26–July 1, 2000 Proceedings, Part II*, vol. 1843 of *Lecture Notes in Computer Science*, pp. 751–767, Springer, Berlin, Germany, 2000.
- [7] K. Kim, T. H. 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.
- [8] O. Barnich and M. Van Droogenbroeck, "ViBe: a universal background subtraction algorithm for video sequences," *IEEE*

Transactions on Image Processing, vol. 20, no. 6, pp. 1709–1724, 2011.

- [9] Y. Wang, D. H. Wang, and W. Fang, “Automatic node selection and target tracking in wireless camera sensor networks,” *Computers & Electrical Engineering*, vol. 40, no. 2, pp. 484–493, 2014.
- [10] Y. Shen, W. Hu, J. Liu, M. Yang, B. Wei, and C. T. Chou, “Efficient background subtraction for real-time tracking in embedded camera networks,” in *Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems (SenSys '12)*, pp. 295–308, Toronto, Canada, November 2012.
- [11] V. Cevher, A. Sankaranarayanan, M. Duarte, D. Reddy, R. Baraniuk, and R. Chellappa, “Compressive sensing for background subtraction,” in *Proceedings of the European Conference on Computer Vision (ECCV '08)*, pp. 155–168, Marseille, France, October 2008.
- [12] G. Warnell, D. Reddy, and R. Chellappa, “Adaptive rate compressive sensing for background subtraction,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '12)*, pp. 1477–1480, Kyoto, Japan, March 2012.
- [13] J. Huang, X. Huang, and D. Metaxas, “Learning with dynamic group sparsity,” in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 64–71, jpn, October 2009.
- [14] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.
- [15] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [16] E. J. Candès, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [17] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

