

Moving Object Segmentation by Length-Unconstrained Trajectory Analysis

Qiyu Liao^(✉), BingBing Zhuang, Jingjing Wang, and Nenghai Yu

CAS Key Laboratory of Electromagnetic Space Information,
University of Science and Technology of China, Hefei, China
{fisher8,bbzhuang,kkwang}@mail.ustc.edu.cn
ynh@edu.ustc.cn

Abstract. Background subtraction for moving cameras is an unsolved key problem in intelligent video analysis. Trajectory analysis has demonstrated a significant difference between background and foreground motion model. But under limitation of trajectory-tracking technique, long-term trajectories are hardly dense and well distributed enough, which may cause inaccuracy in boundary discrimination. Addressed to these problems, in this paper we proposed a robust algorithm of “length unconstrained trajectory analysis” (LUCTA), to recapture “invalidated” information of short trajectories. Extensive experiments demonstrate competitive performance of our frame work on both accuracy and time cost.

Keywords: Background subtraction · Moving cameras · Trajectories

1 Introduction

Background subtraction is the very first step in intelligent video analysis. It aims to find independent moving objects in a relatively static scene. In order to achieve the detection of moving objects in video sequences, many previous works [1–3] have been proposed to construct background model to find the motion region. Recently many works [4, 5] applied smoothness and arbitrariness constraints to strengthened the framework, which have achieved success in many situations. however, assuming that the scene is captured by fixed cameras, they fail to accommodate video taken by moving cameras.

Dealing with moving cameras, work [6] presents a method for modeling in a panoramic view, which is based on some literatures on image mosaic techniques for constructing panoramic views (e.g., [7]). But, they all have an assumption that the camera movement only contains rotation about its optical center which can guarantee no significant parallax, which is proved to be unrealistic in engineering applications. Approach in [8] estimates the motion of some key-points in video by finding and tracking feature points, without constructing an explicit

Q. Liao—This work is supported by National Natural Science Foundation of China (NO. 61371192).

background model. Work [9] finds the trajectories of some key-points, and then tries to construct a panorama with affine camera model. There are still some ways achieve their goal by reconstructing a 3-D scene [10,11]. But to obtain a 3-D scene, multi-view shooting is necessary. All of these works only consider the relative motion between two frames, which may cause instability in classification.

Some works, however, is based on long-term trajectory analysis (LTTA). They estimate the dominant background motion to distinguish unmatched foreground motions. Reference [12] combines trajectory analysis with appearance model to distinguish feature points, and then complete bayesian modeling with these points. Instead of simply applying RANSAC, two constraints are reasonably applied to background and foreground, i.e., low rank and group sparsity constraints [13–15]. Work [16] introduced superpixel to improve the segmentation accuracy. Long-term trajectory analysis, however, is not robust enough to handle dynamic or nonrigid scene, because all of these works require that trajectories must be long enough to construct an accurate low-rank model. As a compromise, what we usually do is simply abandon trajectories last less than required, which means that we have to abandon useful information from occluded region (Fig. 1.). Worse still, occlusion usually happens on the boundaries between moving objects and background. As a consequence, such missing information may cause serious inaccuracy of the final result. Different from all previous works, this paper propose a novel framework called “length-unconstrained trajectory analysis” (LUCTA), which make full use of short trajectories without influencing the property of matrix decomposition. In our framework, a stable model of background motion is constructed with long-term analysis, and applied to separate short trajectories. By doing this, the number of available trajectories is significantly raised, and we can easily improve the accuracy of the boundary of moving object in pixel-level labeling.



Fig. 1. Trajectory distribution: Magenta and green points represent for full length ($P = 50$) and short ($P < 50$) trajectories.

2 Methodology

Our background subtraction algorithm takes a raw video sequence as input, and generates a binary labeling at the pixel level. Figure 2 shows our framework. It has three major steps: low-rank modeling, trajectory classification and pixel level labeling. In the first step, we form dense trajectories of featured points (b), and separate them into tow group: full-length and short trajectories. With

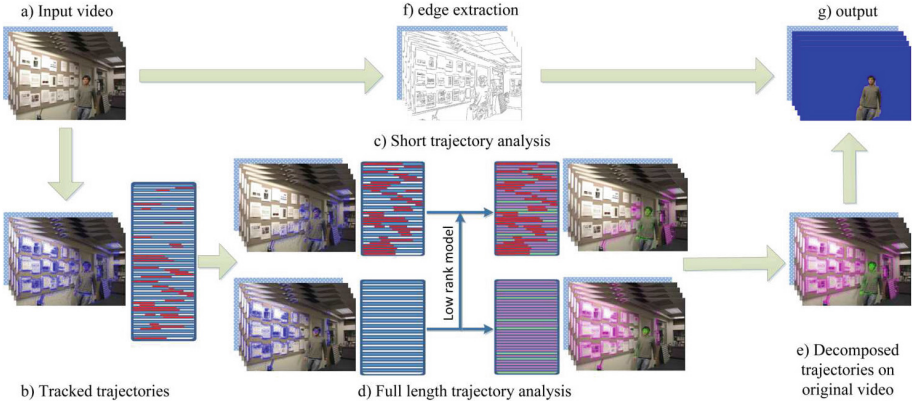


Fig. 2. The framework takes a raw video sequence as input, produces a trajectory matrix (red line is the interrupted parts), analyse this trajectories respectively and produces a binary labeling as the output (Color figure online).

trajectories that pass through all frames, a low-rank motion model is formed (d). With this model, all trajectories are proposed to decomposed into foreground and background (e). Finally, we combine the trajectory information with original graph to produce a pixel level labeling map (g).

2.1 Notations

For a F frame video sequence, P points are tracked. Each trajectory is presented as $p_i = [x_i^1, y_i^1, x_i^2, y_i^2, \dots, x_i^F, y_i^F] \in \mathbb{R}^{1 \times 2F}$, where x and y denote the 2D coordinates in each frame. If the i th point doesn't appear in frame k , x_i^k and y_i^k will be set to 0. The collection of P trajectories can be represented as a $P \times 2F$ matrix, $\Phi = [p_1^T, p_2^T, \dots, p_P^T]^T$, and $\Phi \in \mathbb{R}^{P \times 2F}$. Φ may contain many zeros, because when camera moves, new scene comes into view and old one becomes invisible. Also, trajectory interruption caused by occlusion is another important reason. Similarly, we denote two subset matrices of Φ , Φ_L , presents for trajectories that appear in all F frames, and Φ_S for those less than F . Φ can be decomposed as: $\Phi = B + E$, Where $B, E \in \mathbb{R}^{P \times 2F}$ denote matrices of background and error. Similarly, we have $\Phi_L = B_L + E_L$ and $\Phi_S = B_S + E_S$, where B_L, E_L, B_S, E_S respectively present for background and error matrices of full-length and short trajectories.

2.2 Motion Model of Background

If environment is static, the motion in video depends only on the motion of moving objects, the 3D structure of the scene and the motion of the camera. What we need to do is separate motion information caused by moving object from those caused by camera motion. Trajectories of background tend to be

consistent, while those of foreground are relatively different from primary motion pattern. Generally, full length trajectory matrix B_L can be factored as a $P_L \times 3$ structure matrix of 3D points and a $3 \times 2F$ orthogonal matrix [17]. In other words, background trajectories are proposed to be linearly represented by no more than 3 basis vectors.

Without any prior information, it's not easy to find these basis vectors in a hybrid trajectory matrix. A reasonable solution is to take advantage of low rank and group sparse property. It can be transformed into a convex optimization problem:

$$\begin{aligned} \operatorname{argmin}_{B_L, E_L} \quad & \| \Phi_L - B_L - E_L \|_F^2 \\ \text{s.t. Rank}(B_L) &< 3, \| E_L \| < \alpha k \end{aligned} \quad (1)$$

This equation can be solved by applying two stages iteratively [15]:

- Stage 1: Fix B_L , preserve the αP_L rows with largest values of matrix $\Phi_L - B_L$ as E_L , while the rest rows are set to zero;
- Stage 2: Fix E_L , apply SVD to the matrix $(\Phi_L - E_L) = U\Sigma V^T$, preserve three largest eigenvalues and the rest set zero to form Σ' ; update $B_L = U\Sigma'V^T$.

After convergence, $\Psi = \Sigma'V^T = [\omega_1^T, \omega_2^T, \omega_3^T]$ is the component trajectory matrix we want, where $\omega_1, \omega_2, \omega_3$ represent for 3 component motion of the background.

2.3 Trajectory Level Classification

For full-length trajectories, a fitting function used to establish consensus is the projection error on the three dimensional subspace spanned by the component trajectories. The matrix of component trajectories Ψ is then used to construct a projection matrix, $\Theta = \Psi(\Psi^T\Psi)^{-1}\Psi^T$. By measuring the projection error $f(p_i|\Psi) = \frac{2}{F}\|\Theta p_i - p_i\|_2$, the likelihood that a given trajectory p_i belongs to the background can be easily evaluated.

Short trajectories, however, is not fit for this function. Because some of it's elements is factitiously set to zero, applying such function will lead to huge projection error. Essentially, setting zeros to a trajectory vector is a dimension reducing process. In order to fit for this function, target subspace should have the same dimension. Assuming that p_i is a short trajectory vector, corresponding component trajectories are proposed to be:

$$\omega_j(k)' = \begin{cases} 0 & \text{if } p_i(k) = 0, \\ \omega_j(k) & \text{if } p_i(k) \neq 0. \end{cases} \quad (2)$$

where $k = 1, 2, 3$. Background motion model is presented as $\Psi' = [\omega_1'^T, \omega_2'^T, \omega_3'^T]$, and similarly the low-dimensional projection matrix $\Theta' = \Psi'(\Psi'^T\Psi')^{-1}\Psi'^T$, the likelihood that a short trajectory p_i belongs to the background can be evaluated as $f(p_i|\Psi') = \frac{2}{\|p_i\|_0}\|\Theta' p_i - p_i\|_2$. For convenience, errors under different projection matrices $f(p_i|\Psi')$ and $f(p_i|\Psi')$ will be all named as " $f(p_i)$ " in the remainder

of this paper. Uniform standard can be applied on both full-length and short trajectories: If $f(p_i) > \delta$, p_i is sentenced to foreground trajectory, else sentenced to background trajectory. Given a data matrix of $P \times 2F$ with P trajectories over F frames, the major calculation is $O(PF^2)$ for SVD on each iteration. Considering that we add a short trajectory analysing step, the projection calculation for P' trajectories is $O(PF)$, so the total calculation of the classification is $O(PF^2)$.

2.4 Pixel Level Labeling

The labeled trajectories from the previous step are then used to segment all frames at pixel level. First, the image edge information is abstracted to form an edge map $C(x, y)$ by applying canny operator. For stronger edge detection responses in (x, y) , the lower $C(x, y)$ is. Labels are supposed to be separated on these edges, and minimize the mistakes in calibrations on trajectory level. Now we can transform the labeling question into a combinatorial optimization problem, which minimizes the energy function:

$$\min_{x_p} \left\{ \sum_{p \in G} (V_p(x_p)) + \lambda \sum_{pq \in E} (V_{pq}(x_p, x_q)) \right\} \quad (3)$$

where p represents for pixels in a single frame, and x_p for different labels of these pixels. G is set of tracked points appeared in this frame, while E represents for the neighbouring relation between pixel p and q . Data cost $V_p(x_p)$ can be described by errors of trajectory level calibration, and smooth cost $V_{pq}(x_p, x_q)$ by the edge map. The globally optimal solution can be efficiently computed using graph-cuts [18].

3 Experiments

Given space limitations, this article only lists 4 typical examples to demonstrate the performance of our framework. Three (Vcars, VPerson and VHand) is from video source provided by Sand and Teller, and one (people) is recorded by ourselves. Performances are evaluated by F-Measure: $F = 2recall \times precision / (recall + precision)$. F_T is for trajectory level and F_P is for pixel level. Comparison will mainly be made between results of full-length (LTTA)[15] and proposed length-unconstrained trajectory analysis (LUCTA). Here, ‘‘full-length’’ is defined as a reasonable value of 30 [13], which is neither too long to be tracked nor too short to conduct matrix decomposition; In the other side, ‘‘length-unconstrained’’ means making use of all trajectories lasting between 3 to 50 frames.

3.1 Property of Short Trajectories

In this part, we take video VPerson for example to clarify the effectiveness of short trajectories (Fig. 3). In (a), the amount of trajectories of each length are counted. As shown, number of trajectories lasts more than 50 frames is 6386,

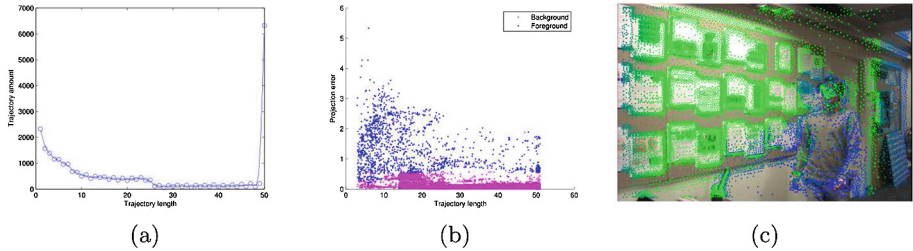


Fig. 3. Performance of short trajectories on test video VPerson: (a). Quantity-length distribution; (b). Projection error-length distribution; (c). Length-position distribution.

while number of short trajectories that lasts between 3 to 49 frames is 20495, most of which (55.2%, 11348) are extremely short (lasts over 3 to 10 frames). Mid-term trajectories only take 16.2% (3388). (b) shows the projection errors distribution in each length. Blue points presents for foreground and magenta ones for background (confirmed by ground truth). The distance of two colors at each column can estimate how well foreground points can be detected. For this sequence, trajectories that lasts more than 27 frames are easily clarified. While length drops to 5 or even 3, although several points may be misjudged, the same threshold is still effective.

In (c) trajectories are set to different colors. If a trajectory is full-length, then it is set to green. And it shades to blue as it's length decreases. Short trajectories often appear on such positions: (1). The edge of the vision; (2). The edges between background and foreground; (3). Nonrigid foreground. We can intuitively tell that (2) and (3) may cause great inaccuracy of the detection.

We also test our framework on three video sequences, and compare classification results. As shown in Fig. 4, been affected by occlusions, there are lots of blank areas in upper lines. By taking short trajectories into consideration, the majority of these areas are made up, which makes description of boundaries between foreground and background more precisely. In our framework, most of short trajectories are correctly classified, and errors are evenly distributed across the frames, which has little influence on the final result.

3.2 Pixel Level Labeling Performance

Since that our work is mainly focused on the contribution of short trajectories, in this part, we will do comparisons between performances with and without applying short trajectories. We also take the two state-of-art algorithms as reference: RANSAC-based [12] and RPCA-based [13] background subtraction. The quantitative and qualitative results are shown in Fig. 5 and Table 1, respectively. Compared with LTTA, LUCTA has lower accuracy but vastly improved quantity (about 3 to 4 times). The reason for decreased accuracy can mainly be attributed as short trajectories often appear around the boundary. Generally, misjudgments of short trajectories often occur as noise points and spread all over the frame.

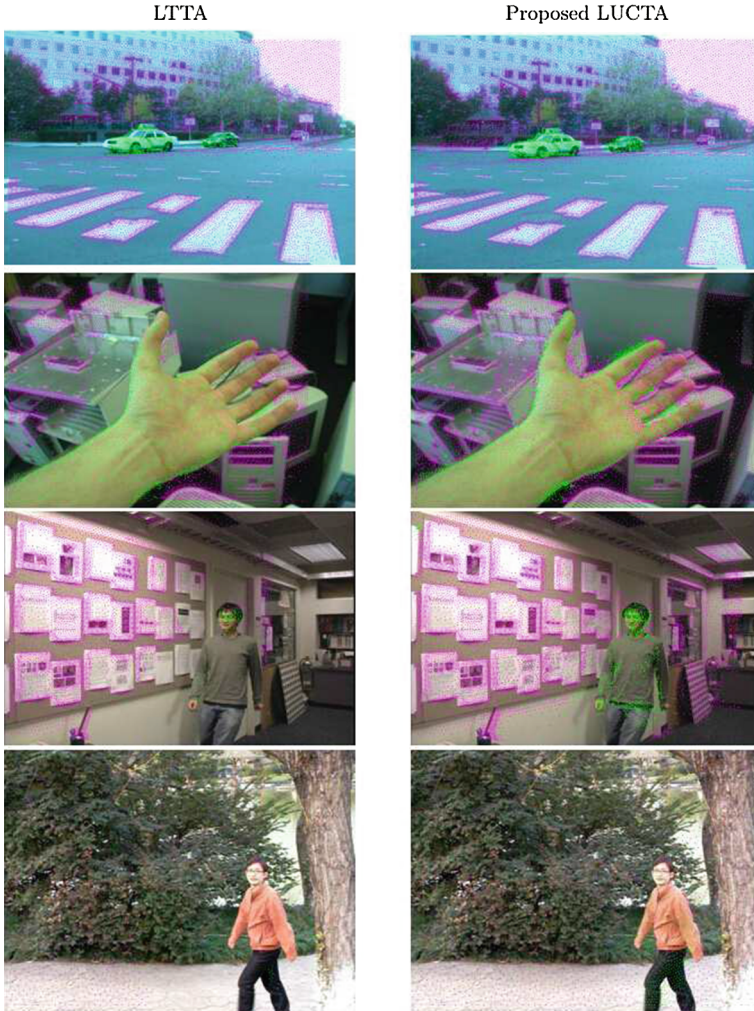


Fig. 4. Performance comparison between LTTA and LUCTA on four video sequences. The upper and below line shows distribution of full length and unconstrained length trajectories in video frames, respectively. By using short trajectories, blank areas caused by occlusion between background and foreground are made up.

In fact, in our experiments we find the effect of such misjudgments on the pixel-level labeling to be small or non-existent. Moreover, although the property of trajectories degenerate as the length decrease, it's still better and more robust than some classic algorithms.

We also compared **time cost** of LTTA and LUCTA. Because the number of 50-length trajectories is much less than number of 30-length ones, the calculation for matrix decomposition is reduced. Taking consideration of time spent on

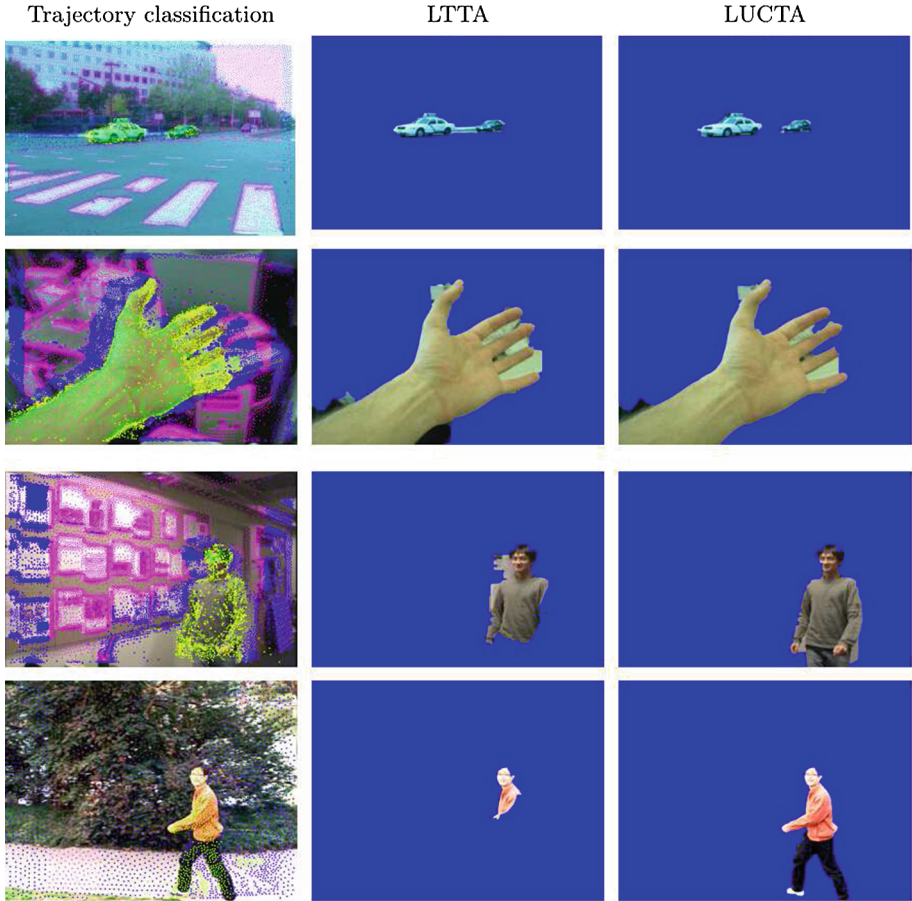


Fig. 5. Result comparison on pixel level labeling: The first column is trajectory classification results. Magenta and blue points represent for detected full-length and short background trajectories; While green and yellow points represent for detected full-length and short foreground trajectories. The second and third column is pixel level labeling results of LTТА and proposed LUCTA (Color figure online).

Table 1. Result comparison

Video	RANSAC F_T	RPCA F_T	LTТА				Proposed LUCTA			
			Number	Time	F_T	F_P	Number	Time	F_T	F_P
VCars	95.5 %	82.5 %	10826	1.41	97.1 %	91.8 %	15506	2.15	95.3 %	98.9 %
VHand	92.0 %	64.5 %	7501	2.99	95.2 %	89.0 %	26789	4.10	86.2 %	95.6 %
VPerson	70.3 %	74.0 %	6386	6.90	93.0 %	73.6 %	23005	5.22	87.6 %	96.5 %
People1	85.2 %	92.4 %	1377	0.88	96.3 %	29.8 %	7242	0.73	87.7 %	94.6 %

Compared with LTТА, our LUCTA shows great advantage in trajectory number, while the accuracy decreases insignificantly.

calculating projection error of short trajectories, in some cases LUCTA is even faster than LTTA, in spite that it's data size is at least 3 to 4 times larger.

4 Conclusion

Because of dimensionality limitation on matrix decomposition process, classical tracking based background subtraction can not acquire accurate boundary. In this paper, we specifically propose a effective method to detect moving regions from video sequences. By using "short trajectories", we get the information of occluded areas, which makes our labeling step much easier. Experiences also demonstrate the competency of our work.

Still and all, we can see that in trajectory classification step, the performance of extreme short trajectories is not satisfying enough. In fact, such misjudgment can be solved by adding constraints of spatial and temporal correlation into optimization step, and we will investigate it in our future work.

References

1. Cristani, M., Bicego, M., Murino, V.: Integrated region-and pixel-based approach to background modelling. In: Workshop on Motion and Video Computing, Proceedings, pp. 3–8. IEEE (2002)
2. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 747–757 (2000)
3. Mittal, A., Huttenlocher, D.: Scene modeling for wide area surveillance and image synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition, Proceedings, vol. 2, pp. 160–167. IEEE (2000)
4. Zhou, X., Yang, C., Weichuan, Y.: Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(3), 597–610 (2013)
5. Guo, X., Wang, X., Yang, L., Cao, X., Ma, Y.: Robust foreground detection using smoothness and arbitrariness constraints. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part VII. LNCS, vol. 8695, pp. 535–550. Springer, Heidelberg (2014)
6. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, vol. 2, pp. II-302–II-309. IEEE (2004)
7. Irani, M., Anandan, P., Hsu, S.: Mosaic based representations of video sequences and their applications. In: Fifth International Conference on Computer Vision, Proceedings, pp. 605–611. IEEE (1995)
8. Sand, P., Teller, S.: Particle video: long-range motion estimation using point trajectories. *Int. J. Comput. Vis.* **80**(1), 72–91 (2008)
9. Jin, Y., Tao, L., Di, H., Rao, N.I., Xu, G.: Background modeling from a free-moving camera by multi-layer homography algorithm. In: 15th IEEE International Conference on Image Processing, ICIP 2008, pp. 1572–1575. IEEE (2008)
10. Taneja, A., Ballan, L., Pollefeys, M.: Modeling dynamic scenes recorded with freely moving cameras. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part III. LNCS, vol. 6494, pp. 613–626. Springer, Heidelberg (2011)

11. Brutzer, S., Hoferlin, B., Heidemann, G.: Evaluation of background subtraction techniques for video surveillance. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1937–1944. IEEE (2011)
12. Sheikh, Y., Javed, O., Kanade, T.: Background subtraction for freely moving cameras. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 1219–1225. IEEE (2009)
13. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? *J. ACM (JACM)* **58**(3), 11 (2011)
14. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc.: Ser. B (Stat. Methodol.)* **68**(1), 49–67 (2006)
15. Cui, X., Huang, J., Zhang, S., Metaxas, D.N.: Background subtraction using low rank and group sparsity constraints. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012, Part I. LNCS*, vol. 7572, pp. 612–625. Springer, Heidelberg (2012)
16. Liao, Q., Liu, B., Yu, N.: Moving object detection for moving cameras on superpixel level. In: *Seventh International Conference on Image and Graphics*, pp. 307–312 (2013)
17. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vis.* **9**(2), 137–154 (1992)
18. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(9), 1124–1137 (2004)