

A New Pixel-Level Background Subtraction Algorithm in Machine Vision

Songsong Zhang, Tian Jiang, Yuanxi Peng^(✉), and Xuefeng Peng

State Key Laboratory of High Performance Computing, College of Computer,
National University of Defense Technology, Changsha 410073, China
13908473894@163.com, {tjiang,pyx}@nudt.edu.cn, pengxf@163.com

Abstract. In the machine vision, the algorithm of background subtraction is used in target detection widely. In this paper, we reproduce some algorithms such as ViBe, Local Binary Similar Pattern (LBSP), Local Ternary Pattern (LTP) and so on. In view of the problem of inaccurate edge in target detection, we propose a method of combining color and LBSP for background model. It can obtain information both in pixel and texture (marked as *improved-LBSP* in the paper). On this basis, we propose a new method marked as *BFs-method* in the paper, which have a new persistence consists of color, LBSP, and time (t). The key advantage of this method lie in its highly robust dictionary model as well as it's ability to automatically adjust pixel-level segmentation behavior, which improves the ability to remove the shadow of the target and the hole inside the target.

Keywords: Background modeling · Machine vision · Background foreground segmentation

1 Introduction

Background subtraction is an important research direction in the field of computer vision. It is widely used in multi-modal video registration, target detection, target tracking. Background modeling is very important for the processing of video, and even directly affects the success or failure of a system. And it can be considered as a classification problem in a certain degree, also known as background foreground segmentation. There are many ways to model the background in background subtraction, including: Single Gaussian (SG) used in real-time tracking of the human body, Mixture of Gaussian Model (MGS) [15], Running Gaussian Average(RGA) [3], CodeBook [8], SOBS-Self-organization background subtraction [17], SACON [19], ViBe algorithm [1], Color [5]. Temporal Median filter [10], W4 and so on. Barnich et al. proposed a visual background extraction (ViBe) algorithm, which just a frame image can established background model, and use the conservative update strategy, the algorithm is simple and real-time. The LBP algorithm uses the imaging principle of human eye to create

a background. There is a trajectory-controlled watershed segmentation algorithm which effectively improves the edge-preserving performance and prevents the over-smooth problem [20].

Our method improved the original LBSP, our main contributions are: (1) combination color and LBSP, which can increase the accuracy of the foreground edge. (2) redefine persistence [12,14], which consists of color, LBSP, and t in time and space respectively, can remove the shadow of the target and the hole inside the target.

2 Relation Work

According to the needs of the project, our method is mainly dealing with static background video, in which the basic idea of the pixel-based model is to establish a background model for each pixel, comparing with the successive frames constantly. Detecting the target based on the difference between the two, and the pixels which is detected as background will be used to update the background model. As time goes on, using the frame to compare with background, updating the background model constantly, and after several frames, the establishment of the background model is stable gradually, and reflect the background more accurately and can adapt to the background scene changes, such as illumination variation.

The general method includes four steps: Pre-processing, Background modeling, Foreground segmentation, and Data verification. At first, pre-processing consists of drying, enhancement, etc. And then background modeling, followed by foreground segmentation. In the process of updating the background model, the segmented data is re-applied to the background model in the form of feedback, complete the background (bg) and foreground (fg) segmentation at last. And there are some models we have learned and improved:

2.1 ViBe

The ViBe algorithm considers background modeling as a classification problem. When the selection is made on the color space, it is judged that the pixel is the foreground pixel by comparing the value of the pixel at the current position with the value of its neighborhood pixel, v_1, v_2, v_3, v_n represent the value of the pixel at the periphery of each pixel or adjacent frame, and each background pixel is modeled by a collection of background sample values, representation such as Eq. 1:

$$M(x) = \{v_1, v_2, \dots, v_N\}. \quad (1)$$

According to its corresponding model $M(x)$, we can classify a pixel value $v(x)$. The specific approach is as follows:

Firstly, define a sphere $S_R(v(x))$ of radius R centered on $v(x)$. Second, Compare $v(x)$ to the closest values with the $S_R(v(x))$. At last, We use $\#min$ for the threshold, we use $\#$ for the number of the intersections between $M(x)$ and $S_R(v(x))$, see Eq. 2, if $\#$ is larger than or equal to the $\#min$, the pixel value $v(x)$

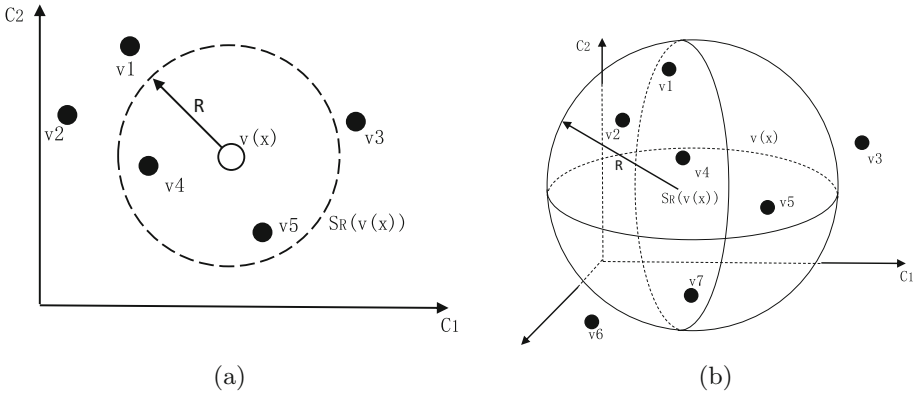


Fig. 1. (a), (b) respectively represent the algorithm in grayscale images and color images

is classified as background, else as foreground. And the algorithm representation respectively in grayscale images and color images as shown in Fig. 1:

$$\# \left\{ S_R(v(x)) \cap \{v_1, v_2, \dots, v_N\} \right\}. \tag{2}$$

2.2 Local Binary Pattern: LBP

One of the significant algorithm of the background subtraction is the LBP [11]. LBP is an algorithm based on feature, which can create a descriptor for each pixel, the specific approach as follows: every pixel as the center pixel to create a 3*3 window, only consider the eight neighbor of a pixel in the basic version, and from the top left corner, clockwise, the eight pixels around the window will be a descriptor of the center pixel, where if the pixel is larger than or equal to center pixel, then marked as 1, else as 0, see Eq. 3:

$$s(x) = \begin{cases} 1 & I_x \geq I_c \\ 0 & I_x < I_c \end{cases} \tag{3}$$

The eight flag bits will produce an eight-bit binary number, which is written down from the upper left corner and clockwise to the binary number. This binary number is converted to a decimal number. The decimal number represents the LBP value of the center pixel. Each pixel will get an LBP according to the neighborhood information through the method. Experiments show that the LBP operator has strong robustness to illumination variation, but the disadvantage is the coverage area is too small to accurately describe the texture requirements.

2.3 Local Binary Similar Pattern: LBSP

The original LBSP is an improvement based on the LBP, and the difference is that it is not a constant that determines whether a pixel (recorded as

observe-pixel in the paper) is a background pixel (recorded as feature *feature-pixel* in the paper), comparing the absolute value of the two instead, see Eqs. 4 and 5.

$$LBSP(x, y) = \sum_{p=0}^{p-1} d(v_p - v_c) \cdot 2^p \quad (4)$$

$$d(v_p - v_c) = \begin{cases} 1 & \text{if } |v_p - v_c| \leq T \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

The advantage of the improvement is that there is a broader and more meaningful range in the background segmentation, which further enhances the robustness of the background modeling algorithm to illumination variation.

3 The Generation of New Methods

Through the study of three background modeling methods, we can find there are many ideas worth learning, the three of them in common is that establishing a background descriptor for each pixel, and comparing the corresponding background pixels with the next frame, except that ViBe is established under the assumption that the background model of a pixel can be described by randomly selecting pixels, and for this reason, the ViBe algorithm does not perform a good segmentation on the initialization if the target in the background model. Both LBP and LBSP use background pixels around a pixel to describe the background, and this description is more representative, the key points of this approach are the selection of the neighborhood and the determination of the threshold. Next, we will introduce our new method detailed.

3.1 Combine Color and LBSP Features to Establish Background Models

The Disadvantage of only Using LBSP Feature Modeling. In the segmentation process of using LBSP, there are some disadvantages of only using LBSP feature modeling: In the segmentation process, the *observe-pixel* segmentation using *feature-pixel* background model is to compare the pixels around *observe-pixel* (marked as *observe-pixel* LBSP in the paper) with the background model (marked as *feature-pixel* LBSP in the paper) [2]. Assuming that the intensity value of pixel *i* is 100 in background modeling and the neighborhood pixels are 40, the L1 distance of *feature-pixel* LBSP is 60, the *feature-pixel* LBSP of the pixel is 0000000000000000; if the neighborhood pixels are 160, the L1 distance of the *observe-pixel* LBSP is still 60, and the *observe-pixel* LBSP is 0000000000000000. It is clearly that the segmentation result is wrong because losing the *observe-pixel* own pixel information. In this case, the experiment results get the noise (stain) shown in (e) of Fig. 2.

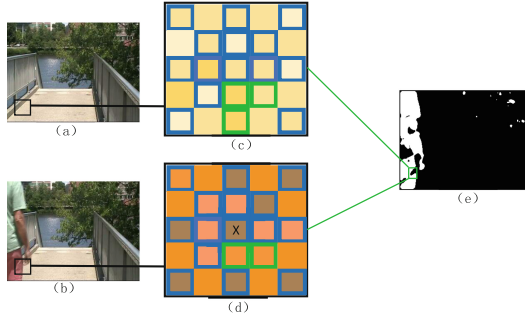


Fig. 2. The drawback of LBSP. (a) Nobody on the road. (b) A pedestrian on the road. (c) The *feature-pixel* LBSP. (d) The *observe-pixel* LBSP. (e) The noise in the results (marked with a green box). (Color figure online)

Combining Color and LBSP Information to Establish the Background Model. In the previous section, we have experimentally demonstrated that the segmentation result is wrong because losing the *observe-pixel* own pixel information. In this part, the LBSP method has been improved (marked as *improved-LBSP* in the paper), combining LBSP and color for background modeling is based on the pixel-level model [13]. We also refer to the color attribute in [9]. The specific process of modeling is taking each pixel as a center, making a 5*5 window, the description includes 25 neighbor pixels and the color value.

The pixel at the center position is marked as v_c . First, the center pixel is described by LBSP, as shown in Eq. 6:

$$LBSP(x, y) = \sum_{p=0}^{p-1} d(v_p - v_c) \cdot 2^p \tag{6}$$

Where v_p is the pixel intensity of the p -th neighbor of v_c on the predefined P -length LBSP pattern. Taking into account the distance between the surrounding pixels and the central pixel are different, in the LBSP character description is treated differently, the specific definition see Eqs. 7 and 8:

$$d(v_p - v_c) = \begin{cases} 1 & \text{if } |v_p - v_c| \leq T \cdot \alpha(p, c) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$\alpha(p, c) = \left(\frac{2 \cdot d(p, c)}{d(p, c) + 1} \right)^2 \tag{8}$$

Where T is the relative LBSP threshold, $\alpha(p,c)$ expresses the different weight due to the difference distance, which is used to determine the threshold, and $d(p, c)$ represents the Euclidean distance between pixels.

Combine the Color and LBSP Information for Pixel Segmentation. In this paper, we have the following conventions: When v_p and v_c belong to the same frame, use *feature-improved-LBSP*. When v_c is the pixel in the background

model, v_p is the pixel around the frame waiting for segmentation, use *observe-improved*-LBSP. The specific process of segmentation is: If the color of the pixel to be segmented is far from the color of the background pixel, it is classified as the foreground directly. Otherwise, the distance between the *observe-improved*-LBSP and the *feature-improved*-LBSP is calculated, and the smaller the Hamming distance the more similar, see Eq. 9, and we will show the process in the next.

$$F(x) = \begin{cases} bg & \text{if } (H(a, b) < T_H) \\ fg & \text{if } (H(a, b) \geq T_H) \end{cases} \quad (9)$$

Algorithm 1. Establish the background model for fg/bg segmentation

```

1: Establish the background model
2: procedure ANY PIXEL:V(X,Y)
3:   if  $|i_{(x,y)observe} - i_c| > T_I$  then
4:     return  $v_{(x,y)}$  is foreground
5:   else
6:     input  $v_p, v_c$ 
7:      $\alpha(p, c) \leftarrow \left(\frac{2-d(p,c)}{d(p,c)+1}\right)^2$ 
8:     if  $|v_p - v_c| \leq T \cdot \alpha(p, c)$  then
9:       return  $d(v_p, v_c) \leftarrow 1$ 
10:    else
11:      return  $d(v_p, v_c) \leftarrow 0$ 
12:    end if
13:     $improved-LBSP_{(x,y)} = \sum_{p=0}^{p-1} d(v_p, v_c) \cdot 2^p$ 
14:    if  $H() > T_H$  then
15:      return  $v_{(x,y)}$  is foreground
16:    else
17:      return  $v_{(x,y)}$  is background
18:       $v_{(x,y)}$  will replace any pixel in the background model of  $v_c$  at a certain
      probability
19:    end if
20:  end if
21: end procedure

```

Where $H()$ represents the Hamming distance, a, b respectively represent of *observe-improved*-LBSP of v_p and *feature-improved*-LBSP of v_c , and T_I represents the intensity threshold. $V(X,Y)$ is segmentation pixel, whose color information is $i_{(x,y)observe}$, v_c, i_c respectively represent the central pixel, and its intensity, v_p is the pixel intensity of the p -th neighbor of the v_c . The update strategy draws on the reference [13] and will not repeat them.

3.2 Modeling the Background with Background Feature (BFs)

BFs Establishment and Foreground Segmentation. We propose a new background modeling method based on the modeling of Local Self Similarity

(LSS) [16] and codebook model [4]. We refer to the expression method in reference [12], and changed the conditions for establishing the background model. For each background pixel, we simply store the data from its surroundings LBSP information derived from the pixel clearly not represent the texture information in the small area, and an LBSP feature is established for each pixel, according to the mapping relation of the position, the eight neighborhood LBSP signatures of each position are taken as the position feature of the pixel, this feature can reflect the central pixel of the local neighborhood of the texture features. In addition, in the segmentation, taking into account the time and space information, we combine color, LBSP, t information as a persistence index, the number of occurrences of a pixel at time t , when the pixel from the last time. In this paper, the background model of the pixel is presented with the background feature (BF), and the background of the pixel is described as the background feature of the pixel. The degree of good and bad is expressed by persistence. The establishment of BFs for each pixel is shown in Fig. 3:

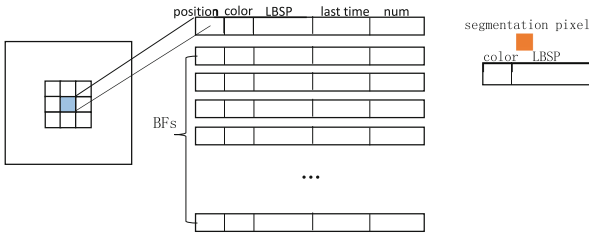


Fig. 3. The process of building BFs

In the pixel segmentation, we first need to calculate the resolution of the pixels to be segmented, that is, the pixels to be split with the corresponding position of the pixels of the similarity of BFs, the expression such as Eqs. 10, 11 and 12:

$$Q_t(x) = \sum_{\omega \in C_L(x)} q(\omega, t) \mid (d(I_t(x), \omega) < D(x)) \tag{10}$$

Where $C_L(x)$ is the current pixels BFs, and ω is a BF, $d(I_t(x), \omega) < D(x)$ indicating that the pixel to be segmented with the ω in the BFs meet the color/LBSP threshold and $Q_t(x)$ is persistence.

$$q(w, t) = \frac{n \cdot \left(p \cdot \alpha + \frac{1}{t-t_{last}} \cdot \beta \right)}{S_{num}} \tag{11}$$

Where n is the number of occurrences of w in BFs, p is the position information of w , α is the position weight, t_{last} is the time of the last occurrence of w , β is the time weight, S_{num} is the total number, who is a constant in the experiment.

$$S_t(x) = \begin{cases} 1 & \text{if } Q_t(x) < W(x) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Where $W(x)$ represents the threshold of the background pixel persistence, and if the pixel is divided into a foreground, the result is 1; otherwise, the result is 0. $D(x)$ and $W(x)$ are dynamically adjusted, and the result of the segmentation is fed back to the BFs for the next segmentation [12].

Update Strategy. We compare the colors through L1 and compare the LBSP by the Hamming distance when calculates persistence. Combination the three information can improve the robustness to illumination variation. When the light of the observation scene has changed, we need to modify a small part of the BFs immediately. Only when the color/LBSP distortion is very small, we make random updates. If the change is only temporary, the small part that has not changed will restore the model to its original state quickly. We also draw on the sampling consistency model mentioned in [7, 18], that is, if a pixel is divided into a background, the neighboring pixel models are replaced as their local expression. This update strategy is different from the improvement of the spatial background mentioned in codebook [11], which is more predictable because it can be carried out under the influence of external conditions such as camera shake [12].

4 Experiments and Conclusions

The experimental platform is Intel(R) Core(TM) i5-4570, CPU@3.20 GHz, RAM 8 GB, the development tool of the algorithm is OpenCV2.4.10 + VS2010.

4.1 Experiment

Experiment1: In the improved part of LBSP described in 3.1, we used the data set CDnet dataset [6]. This data set contains a rich set of video sequences. We selected one of these data for experimentation, and there are qualitative and quantitative analysis of the experimental results. We refer to the experimental parameters in [2], and the experimental parameters we use are: $T_H = 4$, $T = 30$, $T_I = 30$. The results are shown in Fig. 4:

In fact, we also change the size of the parameters in the experiment, $T_H = 1$ (low), 28, 50(high) and with $T = 3$ (low), 30, 90(high). And we will evaluate the model quantitatively from the following six index mentioned in [11].

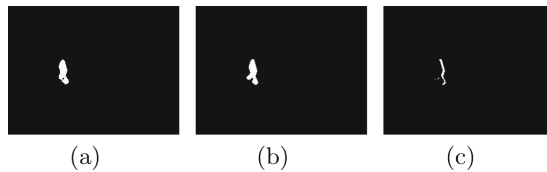


Fig. 4. (a) The results of the reference [1] (b) The results of this method in 3.1 (c) The difference between the two (edge information)

- (1) Recall (Re): $TP / (TP + FN)$
- (2) Specificity (Sp): $TN / (TN + FP)$
- (3) False Positive Rate (FPR): $FP / (FP + TN)$
- (4) False Negative Rate (FNR): $FN / (TN + FP)$
- (5) Precision (Pr): $TP / (TP + FP)$
- (6) F-measure: $2 * Pr * Re / (Pr + Re)$

Where TP is the number of pixels at the correct location, TN is the number of true negatives, FN is number of false negatives, and FP is the number of pixels at the wrong location.

It can be seen T has a greater effect on the results from the experimental results. When T is small, the model is not robust for the noise. When T becomes larger, our model is more robust for the noise. The experimental results are shown in Table 1 and Fig. 5:

Table 1. The value of the index under different thresholds

Threshold	Index					
	Recall	Specifity	FPR	FNR	Precision	F-Measure
$T_H = 1, T = 3$	0.9841517331	0.9603088139	0.0396911861	0.0158482669	0.1975477899	0.3290464217
$T_H = 1, T = 30$	0.9250248188	0.9996272409	0.0003727591	0.0749751812	0.9609958018	0.9426672834
$T_H = 1, T = 90$	0.8704055379	0.9996680584	0.0003319416	0.1295944621	0.9630098738	0.9143690207
$T_H = 28, T = 3$	0.9837537377	0.9830047332	0.0169952668	0.0162462623	0.3649595923	0.5324042627
$T_H = 28, T = 30$	0.8557587104	0.9998242836	0.0001757164	0.1442412896	0.9797378360	0.9135611709
$T_H = 28, T = 90$	0.3790183376	0.9999999408	0.0000000592	0.6209816624	0.9999842689	0.5496905857
$T_H = 50, T = 3$	0.9837537377	0.9830047332	0.0169952668	0.0162462623	0.3649595923	0.5324042627
$T_H = 50, T = 30$	0.8557587104	0.9998242836	0.0001757164	0.1442412896	0.9797378360	0.9135611709
$T_H = 50, T = 90$	0.3790183376	0.9999999408	0.0000000592	0.6209816624	0.9999842689	0.5496905857

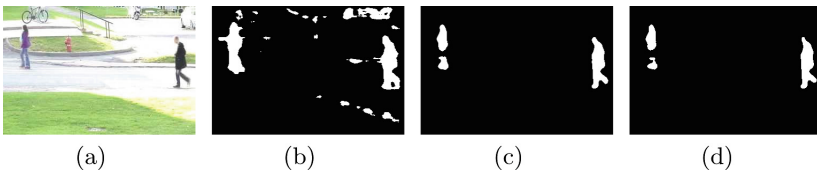


Fig. 5. (a) The original image in the dataset (b) Background subtraction result, when $T_H = 1, T = 3$ (c) Background subtraction result, when $T_H = 1, T = 30$ (d) Background subtraction result, when $T_H = 1, T = 90$

Experiment2: We also used the dataset CDnet dataset [6] to model the BFs. In the experiment, the number of BFs is fixed, the default is 50. In the static background, each background model can do a good split only need to activate two or three BFs. In the dynamic background, up to 20 BFs is activated. From the experimental results, our method can make the target complete which is blocked and camouflaged. Our improved method is denoted as BFs-method, the experimental results are shown in Fig. 6:

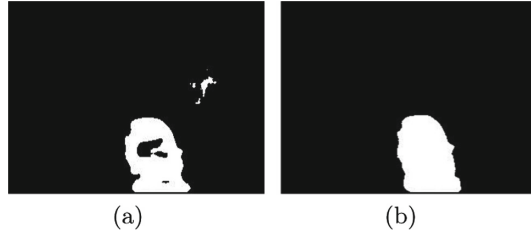


Fig. 6. (a) Experimental result of [12] (b) Experimental result of BF-method

We also used the quantitative assessment in the previous section. The results are shown in Table 2. In this table, it shows that the BF-method has a higher value in Recall, and has a good result in Precision (Pr). In addition, FPR has decreased compared with the other six methods, and FNR is lower than PAWCS. This result shows that the persistence consists of color, LBSP, t can reflect the characteristics of the pixel background better, of which the number of occurrences can reflect the life of a pixel. In addition, the segmentation results clearly exceed the *improved-LBSP* which is described in Sect. 3.1.

Table 2. Quantitative assessment of several methods

Method	Index					
	Recall	Specifity	FPR	FNR	Precision	F-Measure
PBAS	0.78	0.990	0.010	0.009	0.82	0.75
ViBe+	0.69	0.993	0.007	0.017	0.83	0.72
SC-SOBS	0.80	0.983	0.017	0.009	0.73	0.73
LOBSTER	0.81	0.982	0.019	0.010	0.77	0.75
KDE	0.65	0.993	0.007	0.025	0.77	0.64
GMM	0.51	0.995	0.005	0.029	0.82	0.59
LSS	0.94	0.851	0.149	0.002	0.41	0.51
ED	0.71	0.969	0.031	0.017	0.62	0.61
PAWCS	0.85	0.991	0.050	0.140	0.87	0.85
<i>Improved-LBSP</i>	0.88	0.999	0.001	0.120	0.13	0.92
BFs-method	0.93	0.999	0.001	0.075	0.96	0.94

4.2 Conclusion

We change the condition of establishing LBSP by adding the weight of distance based on [1]. From the experimental results, we can see that our method can get more accurate edge of the targets and remove the noise in the background when we set suitable value of parameters. In addition, BF-method combines

LBSP, color, persistence to build BF's, in which persistence is an index of the classification of background words, because it combines information of time and space, so it is superior to the classification of background words. BF's-method can remove the shadow of the target and the hole inside the target.

Acknowledgements. This work is supported by the Research Fund of State Key Laboratory of High Performance Computing under Grant No. 201612-01, National University of Defense Technology.

References

1. Barnich, O., Droogenbroeck, M.V.: Vibe: a universal background subtraction algorithm for video sequences. *IEEE Trans. Image Process.* **20**(6), 1709–24 (2011). A Publication of the IEEE Signal Processing Society
2. Bilodeau, G.A., Jodoin, J.P., Saunier, N.: Change detection in feature space using local binary similarity patterns. In: *International Conference on Computer and Robot Vision*, pp. 106–112 (2013)
3. Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking (2016)
4. Fan, Z., Chen, X., Chen, L.: Improvement strategy based on brightness ranges in codebook algorithms. *J. China Univ. Metrol.* (2013)
5. Fernndez, E., Besuievsky, G.: Efficient inverse lighting: a statistical approach. *Autom. Constr.* **37**(37), 48–57 (2014)
6. Goyette, N., Jodoin, P., Porikli, F., Konrad, J.: Changedetection.net: a new change detection benchmark dataset. In: *Computer Vision and Pattern Recognition Workshops*, pp. 1–8 (2012)
7. Hofmann, M., Tiefenbacher, P., Rigoll, G.: Background segmentation with feedback: the pixel-based adaptive segmenter. In: *Computer Vision and Pattern Recognition Workshops*, pp. 38–43 (2012)
8. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground-background segmentation using codebook model. *Real-Time Imaging* **11**(3), 172–185 (2005)
9. Liang, Z., Liu, X., Liu, H., Chen, W.: A refinement framework for background subtraction based on color and depth data. In: *IEEE International Conference on Image Processing*, pp. 271–275 (2016)
10. Lo, B.P.L., Velastin, S.A.: Automatic congestion detection system for underground platforms. In: *International Symposium on Intelligent Multimedia, Video and Speech Processing*, pp. 158–161 (2001)
11. Nonaka, Y., Shimada, A., Nagahara, H., Taniguchi, R.: Evaluation report of integrated background modeling based on spatio-temporal features. In: *Computer Vision and Pattern Recognition Workshops*, pp. 9–14 (2012)
12. St-Charles, P.L., Bilodeau, G.A., Bergevin, R.: A self-adjusting approach to change detection based on background word consensus. In: *IEEE Winter Conference on Applications of Computer Vision*, pp. 990–997 (2015)
13. Stcharles, P.L., Bilodeau, G.A.: Improving background subtraction using local binary similarity patterns. In: *Applications of Computer Vision*, pp. 509–515 (2014)
14. Stcharles, P.L., Bilodeau, G.A., Bergevin, R.: Online multimodal video registration based on shape matching. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 26–34 (2015)

15. Suo, P., Wang, Y.: An improved adaptive background modeling algorithm based on Gaussian mixture model. In: International Conference on Signal Processing, pp. 1436–1439 (2008)
16. Torabi, A., Bilodeau, G.A.: Local self-similarity-based registration of human rois in pairs of stereo thermal-visible videos. *Pattern Recogn.* **46**(2), 578–589 (2013)
17. Van Droogenbroeck, M., Barnich, O.: Visual background extractor (2011)
18. Van Droogenbroeck, M., Paquot, O.: Background subtraction: experiments and improvements for vibe. In: Computer Vision and Pattern Recognition Workshops, pp. 32–37 (2012)
19. Wang, H., Suter, D.: Background subtraction based on a robust consensus method. In: International Conference on Pattern Recognition, pp. 223–226 (2006)
20. Yin, X., Wang, B., Li, W., Liu, Y., Zhang, M.: Background subtraction for moving cameras based on trajectory-controlled segmentation and label inference. *KSII Trans. Internet Inf. Syst.* **9**(10), 4092–4107 (2015)