A Robust Local Binary Similarity Pattern for Foreground Object Detection

Dongdong Zeng, Ming Zhu and Hang Yang

Introduction: Foreground object detection for a stationary camera is one of the essential tasks in many computer vision and video analysis applications such as object tracking, activity recognition, and human-computer interactions. As the first step in these high-level operations, the accurate extraction of foreground objects directly affects the subsequent operations. Background subtraction is the most popular technology used for foreground object detection, the performance of foreground extract highly depends on the reliability of background modeling. In the past decades, various background subtraction methods have been proposed by researchers, most of them are classified as pixel-based method due to their low complexity and high processing speed. However, pixel-based methods are very sensitive to illumination variations in the scene, and each comparison will produce a two-bits encoder. However, recent

research shows that the two-bits encoding is a coarse thresholding scheme due to the non-normal distribution of background statistics. Calculating the feature should not be based on strictly comparisons such as < and >, but using absolute difference. Thus, in [4], Local Binary Similarity Pattern (LBSP) operator proposes to calculate local similarity in a 5 × 5 region based on the absolute difference. Fig. 1(a–c) presents the encoding pattern of the three described texture operators.

However, these LBP-like features are often too sensitive to local changes in dynamic background and highly noise regions. To solve this problem, we propose a robust local binary similarity pattern operator called RLBSP. Let a pixel p at a certain location of an image I, the coordinate of p is (xp, yp) and there are P neighboring subregions gi, spanned on an n × n region R (see Fig. 1(d)). Then the RLBSP operator applied to p can be expressed as follows:

\[ RLBSP_{p,p}(p) = \bigoplus_{i=0}^{P-1} S(I_{g_i}, I_p) \]

where \( I_p \) is the intensity value of center pixel \( p \), \( I_{g_i} \) is the average intensity value of its neighboring subregion \( g_i \), \( \bigoplus \) denotes concatenation operator of binary strings and \( S \) is a thresholding function which is defined as:

\[ S(I_{g_i}, I_p) = \begin{cases} 1, & \text{if } |I_{g_i} - I_p| \leq \tau I_p, \\ 0, & \text{otherwise.} \end{cases} \]

where \( \tau \) is a relative threshold value (which is set to 0.14 in this paper).

The neighboring sub-areas \( g_i \) are calculated as follows:

\[ g_0 = (g_0 + g_7 + r_3 + r_7) / 4; g_8 = (r_2 + g_3 + r_4 + r_9 + r_10 + r_11) / 6; \\
 g_1 = (r_5 + r_6 + g_12 + r_13) / 4; g_9 = (g_0 + r_10 + r_11 + r_12 + r_13) / 6; \\
 g_2 = (r_7 + r_8 + g_14 + r_15) / 4; g_{10} = (r_4 + r_12 + r_13 + r_22 + r_23 + r_25 + r_26) / 6; \\
 g_3 = (r_11 + r_7 + r_20 + r_23) / 4; g_{11} = (r_15 + r_16 + r_22 + r_23 + r_24 + r_25 + r_26 + r_27) / 6; \\
 g_4 = (r_23 + r_24 + r_26 + r_29) / 4; g_{12} = (r_18 + r_19 + r_24 + r_25 + r_31 + r_32) / 6; \\
 g_5 = (r_22 + r_23 + r_30 + r_40 + r_41 + r_32 + r_43 + r_47) / 6; \\
 g_6 = (r_25 + r_34 + r_42 + r_43) / 4; g_{13} = (r_29 + r_30 + r_31 + r_32 + r_33 + r_34 + r_43) / 6; \\
 g_7 = (g_0 + r_6 + r_27 + r_28) / 4; g_{14} = (r_24 + r_30 + r_31 + r_32 + r_43 + r_44 + r_45) / 6. \]

Robust Local Binary Similarity Pattern: Binary feature operators are popularly employed in background subtraction area due to their low complexity, discriminative and illumination invariance. Most of them are based on the comparisons between pixel pairs in different configurations. For example, Local Binary Pattern (LBP) works with eight-neighbors of each pixel, using the value of the center pixel as the threshold and considering the result as a binary string. The Scale Invariant Local Ternary Pattern (SILTLP) operator defines a new threshold strategy which makes the operator invariant under scale transform of pixel intensities, and each comparison will produce a two-bits encoder. However, recent

Fig. 1 Examples of LBP [1], SILTP [2], LBSP [3] and the proposed robust local binary similarity pattern (RLBSP).
this letter, we proposed to create our background model by integrating advantages of both texture feature and color feature. Since texture features are not numerical values, traditional background modeling methods such as GMM [6], KED [7] are not suitable for modeling RLBSP into background. Fortunately, inspired by the sample consensus methods such as GMM [6], KED [7] are not suitable for modeling features are not numerical values, traditional background modeling advantages of both texture feature and color feature. Since texture this letter, we proposed to create our background model by integrating

Algorithm 1: Foreground Object Detection Procedure

1: Initialization.
for each pixel \( p(x, y) \) in the first frame \( I^0 \) do
extract color feature \( I(x, y) \) and texture feature \( RLBSP(x, y) \)
construct the background sample of \( p(x, y) \) with:
\[
F(x, y) = \{ \bar{I}(x, y), RLBSP(x, y) \}
\]
construct the background model of \( p(x, y) \) with \( N \) random neighboring background samples:
for \( i = 1, \ldots, N \) do
\[
B_i(x, y) = \{ \bar{I}(\bar{x}, \bar{y}) | (\bar{x}, \bar{y}) \in \N(x, y) \}
\]
end for
end for

2: Foreground Detection.
for each pixel \( p(x, y) \) in the current frame \( I \) do
extract color feature \( I(x, y) \) and texture feature \( RLBSP(x, y) \)
\( n_{\text{Matches}} = 0, i = 0 \)
while \( n_{\text{Matches}} < \#_{\text{min}} \) and \( i < N \) do
\[
\text{colorDist} = L_1 \text{Dist}(I(x, y), \bar{I}(x, y))
\]
if \( \text{colorDist} \geq R_c \) goto failedMatch;
\[
\text{textDist} = \text{HamDist}(RLBSP(x, y), RLBSP_i(x, y))
\]
if \( \text{textDist} \geq R_t \) goto failedMatch;
\( n_{\text{Matches}}++ \);
failedMatch: 
\( i++ \);
end while
if \( n_{\text{Matches}} < \#_{\text{min}} \) then
\( p(x, y) \) is foreground;
else
\( p(x, y) \) is background;
end for

3: Background Model Update.
for each pixel \( p(x, y) \) in the current frame \( I \) do
if \( p(x, y) \) is background
\[
\text{rand} \% \phi = 0
\]
update \( B(x, y) \) with \( F(x, y) \)
if \( \text{rand} \% \phi = 0 \)
update \( B(x, y) \) with \( F(x, y) \)
end for

Experimental results: To evaluate the use of RLBSP for foreground object detection, we use a standard Change Detection dataset (CDNet 2012) available in [5]. Seven different quantitative metrics have been defined: Recall (Re), Specificity (Sp), False positive rate (FPR), False negative rate (FNR), Percentage of wrong classifications (PWC), Precision (Pr), F-Measure (FM). All these metrics range from 0 to 1, for Re, Sp, Pr and FM metrics, higher values indicate more accuracy while for PWC, FNR and FPR metrics, lower values indicate better performance. Generally speaking, a foreground detection method is considered good if it gets high recall scores and without sacrificing precision. So, the F-Measure metric is mainly accepted as a good indicator of the overall performance of different methods. We use the tools provided by the authors of [5] to compute these metrics. In Table 1 we report the seven metric scores on each category in CDNet 2012 dataset. We can see that our method performs well on the dynamic background, camera jitter and shadow categories. Then, we compare the F-Measure performance of the proposed method against some state-of-the-art methods in Table 2, the results show that our RLBSP-based foreground detection method is much more robust than other methods with LBSP [4] and texture features on the dynamic background and

### Table 1: Complete results obtained by RLBSP on the CDNet 2012 dataset

| Category         | Recall | Spicity | FPR  | FNR  | PWC  | Precision | F-Measure |
|------------------|--------|---------|------|------|------|-----------|-----------|
| baseline         | 0.9409 | 0.9973  | 0.0027 | 0.0591 | 0.4842 | 0.9144    | 0.9272    |
| cameraJ          | 0.8578 | 0.9775  | 0.0225 | 0.1122 | 2.6135 | 0.7290    | 0.7863    |
| dynamic          | 0.9068 | 0.9941  | 0.0059 | 0.0932 | 0.6788 | 0.6754    | 0.7185    |
| intermittent     | 0.7065 | 0.9294  | 0.0706 | 0.2935 | 4.8502 | 0.6263    | 0.6138    |
| shadow           | 0.9543 | 0.9911  | 0.0009 | 0.0657 | 1.1257 | 0.8437    | 0.8850    |
| thermal          | 0.7863 | 0.9945  | 0.0055 | 0.2137 | 1.3863 | 0.8748    | 0.8117    |
| Overall          | 0.8603 | 0.9807  | 0.0193 | 0.1196 | 2.4564 | 0.7773    | 0.7904    |

### Table 2: Per-category and overall F-Measure scores by different methods

| Method      | Baseline | CameraJ | Dynamic | Intermitent | Shadow | Thermal | Overall |
|-------------|----------|---------|---------|-------------|--------|---------|---------|
| RLBSP       | 0.9272   | 0.7663  | 0.7185  | 0.6138      | 0.8550 | 0.3117  | 0.5904  |
| LBSP        | 0.9320   | 0.7462  | 0.5664  | 0.5094      | 0.8696 | 0.7803  | 0.7481  |
| KED         | 0.9004   | 0.7311  | 0.6278  | 0.4816      | 0.8099 | 0.6331  | 0.6993  |
| GMM         | 0.9520   | 0.7250  | 0.7841  | 0.7434      | 0.7600 | 0.7423  | 0.6719  |

camera jitter categories. The processing speed of the proposed method is also evaluated to show that it has low computational complexity and is suitable for real time applications. For a sequence with the frame size of 320 × 240, it runs at 41 fps with a 3.6 GHz Intel Core i7 7700 CPU.

Conclusion: In this paper, we propose a robust texture operator named Robust Local Binary Similarity Pattern (RLBSP) for foreground object detection. The RLBSP shows strong robustness to illumination variations and dynamic backgrounds scene conditions. To handle the limitation of texture feature in uniform regions, we combine the color and texture features to characterize pixel representations. Experiments evaluated on the CDNet 2012 dataset show that the proposed method outperforms state-of-the-art methods and runs in real-time performance.

Acknowledgment: This research is supported by the National Science Foundation of China under Grant No.61401425.

E-mail: zengdongdong13@mails.ucas.edu.cn

References
1. M. Heikkila and M. Pietikainen, “A texture-based method for modeling the background and detecting moving objects,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 4, pp. 657–662, 2006.
2. X. Tan and B. Triggs, “Enhanced local texture feature sets for face recognition under difficult lighting conditions,” IEEE Trans. Image Process., vol. 19, no. 6, pp. 1635–1650, 2010.
3. S. Liao, G. Zhao, V. Kellokumpu, M. Pietikäinen, and S. Z. Li, “Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2010., pp. 1301–1306.
4. P-L. Ste-Charles and G.-A. Bilodeau, “Improving background subtraction using local binary similarity patterns,” in Proc. IEEE Winter Conf. Comput. Vision., 2014., pp. 509–515.
5. N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, “Changendetection.net: A new change detection benchmark dataset,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, 2012., pp. 1–8.
6. C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 1999., pp. 246–252.
7. A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance,” Proc. IEEE., vol. 90, no. 7, pp. 1151–1163, 2002.
8. O. Barnich and M. Van Droogenbroeck, “VBoe: A universal background subtraction algorithm for video sequences,” IEEE Trans. Image Process., vol. 20, no. 6, pp. 1709–1724, 2011.
9. J. Yao and J.-M. Odobez, “Multi-layer background subtraction based on color and texture,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2007., pp. 1–8.