Improving Background Subtraction using Local Binary Similarity Patterns

Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau
LITIV lab., Dept. of Computer & Software Eng.
École Polytechnique de Montréal
Montréal, QC, Canada
{pierre-luc.st-charles,gabilodeau}@polymtl.ca

Abstract

Most of the recently published background subtraction methods can still be classified as pixel-based, as most of their analysis is still only done using pixel-by-pixel comparisons. Few others might be regarded as spatial-based (or even spatiotemporal-based) methods, as they take into account the neighborhood of each analyzed pixel. Although the latter types can be viewed as improvements in many cases, most of the methods that have been proposed so far suffer in complexity, processing speed, and/or versatility when compared to their simpler pixel-based counterparts. In this paper, we present an adaptive background subtraction method, derived from the low-cost and highly efficient ViBe method, which uses a spatiotemporal binary similarity descriptor instead of simply relying on pixel intensities as its core component. We then test this method on multiple video sequences and show that by only replacing the core component of a pixel-based method it is possible to dramatically improve its overall performance while keeping memory usage, complexity and speed at acceptable levels for online applications.

1. Introduction

Background subtraction based on change detection is the first step used in many video analysis applications to detect and segment foreground objects. Hence, the end results quality often depends largely on the method used. A naive approach based on directly comparing pixel intensities over multiple video frames would not be practical in most cases, since real video sequences often contain outliers in the form of noise and dynamic elements (varying illumination, undulating objects, etc.) that should not be considered as foreground. Consequently, more precise and complex reference models and methods have been developed over the years to fill the need for robust applications.

What we propose in this paper is an "all-in-one", single model, single update scheme, training-less, *spatiotemporal-based* background subtraction method which is in part derived from the non-parametric ViBe method[1]. The key aspect to spatial and temporal neighborhood analysis here is that instead of solely using pixel-level intensities as the core component to create our reference model (in a ViBe-like paradigm), we complement it with the output of a modified Local Binary Similarity Pattern (LBSP) descriptor[3]. In other words, we show that even without modifying a generic method's initialization, maintenance and foreground detection rules, it is possible to achieve proper *spatiotemporal* analysis by only incorporating a minimalistic feature-based component in its reference model. We also present in the paper a few general improvements that were implemented in our method, coined LOBSTER (LOcal Binary Similarity segmenTER), over a more "straightforward" integration strategy.

Altogether, using the proposed approach, standardized evaluations using the recently-published CDNet dataset[7] demonstrate that we extract enough spatial and temporal neighborhood information in video sequences to dramatically increase the base algorithm's performance across multiple complex scenarios, without compromising baseline results. Also, we manage to keep extra memory, processing time and post-processing operations to a minimum while outperforming other much more complex state-of-the-art methods also tested in [7], including ViBe+[5].

2. Related Work

The most simple and earliest examples of change detection algorithms are the ones based on the idea that the background can be modeled by independently observing every pixel's intensity subsequently (*i.e.* the so-called *pixel-based* algorithms) in order to then determine if a given intensity value is a true outlier (to be considered foreground). Some of these methods, such as the ones based on density estimation (via parametric Mixture of Gaussians[23] or non-parametric Kernel Density Estimation[6]) are now extremely well-known and have been the root of countless variations, improvements and extensions over the years (*e.g.* [17, 30]). Other methods have also been developed using different perspectives in order to reach the same goal: for example, ViBe[1] and PBAS[9] are both *pixel-based* methods according to their ideology, but they differ from the more traditional approaches as they base their models on random pixel sampling and label diffusion.

Collectively, although *pixel-based* methods are generally lightweight, effective and easy to bootstrap, they ignore part of the information present in video sequences, such as spatial relations between pixels. *Spatial-based* methods (essentially similar to *region-based*, *texture-based* or *block-based* methods), on the other hand, attempt to harness this information using, for example, feature or block descriptors[8, 4, 25], motion-based analysis[10] and even local color histograms[13] in order to strengthen their models. Typical cases where this ideology is beneficial include foreground occlusions with pixel intensities that are similar to the background's and global illumination changes (where textures are unaffected).

Subsequently, *temporal* and *spatiotemporal-based* methods attempted to surpass previous work by taking into account temporally repetitive behavior between the model and the preceding, current and sometimes even upcoming frames of the analyzed sequence. Such information can primarily help with short-term intensity changes, as what is commonly seen in dynamic backgrounds (rippling water, swaying trees, *etc*.). Such a solution is presented in [22], and is based on the bidirectional temporal analysis and matching of pixel intensities variations.

There have been numerous attempts to merge the advantages of different model categories by creating change detection algorithms that rely on multiple analysis techniques simultaneously[24], post-processing label validation[10, 22, 26] and/or advanced morphological operations[5]. However, while all these approaches are generally very successful at improving the results of their base algorithm, they often suffer from highly increased processing (and/or post-processing) time, and sometimes even require an offline training or pre-analysis phase that is not always suitable for real-time applications.

As for the use of local binary patterns and their derivatives in background subtraction, the work of Heikkilä and Pietikäinen in [8] introduces one of the first examples of reliable and efficient *texture-based* methods in the domain. Since then, many contenders have come up with alternatives: Yoshinaga *et al*. [28], for example, present an integration of both *spatial-based* and *pixel-based* methodologies using MoGs over different pixel pairs defined by a modified LBP pattern. Zhang *et al*. [29], on the other hand, propose to combine spatial texture and temporal motion using weighted LBP histograms. More recently, a new type of local binary pattern based on self-similarities throughout images (LBSP) was proposed in [3] and was briefly used to demonstrate its efficiency against *pixel-based* methods in baseline scenarios.

3. Methodology

As stated earlier, our proposed background subtraction method is based on the adaptation and integration of the LBSP features to a non-parametric technique's set of model building and maintenance rules. We chose to start off from the original ViBe approach because it offered the prospect of a flexible and lightweight method that left plenty of breathing space for future improvements. Also, to the author's knowledge, no earlier attempts to adapt ViBe to use a feature-based reference model had been proposed. Finally, it also offered a very simple way to compare our results to a pure pixel-based method without having to stray too far from its original implementation.

Thus, the straightforward approach for this adaptation would have been to simply replace all pixel intensity-related concepts by their feature descriptor-based counterparts. In the case of ViBe, the method would still be based on a reference model that uses N background samples per pixel to independently determine which new pixels are outliers (foreground) : the difference would reside in the nature of these samples, which would be replaced by LBSP binary string descriptors, and in the method used for comparisons, which would become a Hamming distance operator. In reality, although this solution is feasible (and its overall results will be presented in section 4.2), we propose multiple low-cost improvements to both the model's rules and the feature descriptor in order to maximize component synergy.

3.1. Pixel Characterization Model

The feature descriptor proposed in [3] is quite easy to understand: using a small predetermined pattern, a binary vector is constructed from the comparisons of pixel intensities centered around a point of interest. Unlike a traditional LBP approach which calculates the difference between two values, the LBSP approach returns whether they are similar or not (via absolute difference). The key to its temporal component (and sensitivity to illumination variation) lies in its ability to use the central pixel intensity of a previous frame for new comparisons (referred to as inter-image LBSP by the authors). Hence, one of the disadvantages of this technique is that in order to build a *spatiotemporal* reference model for change detection, we need to keep both feature information and pixel intensity information jointly up to date.

The first improvement we propose for our background model is based on the observation that only relying on LBSP comparisons can sometimes be unsuitable in noisy or blurred regions. Such a case is illustrated in Figure 1:

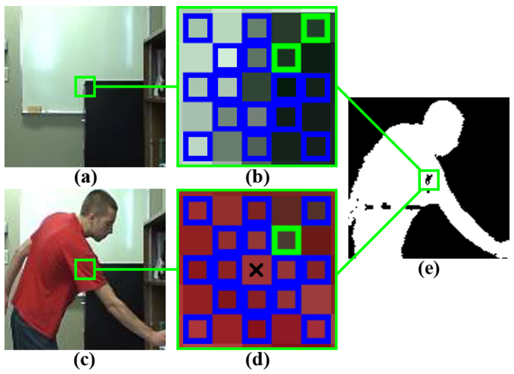

Figure 1. Typical failure case when only LBSP features are used for change detection on CDNet's office sequence; (a) is the background image recreated from the model, (b) illustrates one of the intra-LBSP descriptors computed in the highlighted region (where dark blue squares indicate an intensity mismatch with the current central pixel and light green squares a match), (c) is an input frame to be segmented, (d) illustrates an inter-LBSP descriptor computed on the input frame at the same coordinate as (b) and using its central pixel intensity as reference, and (e) shows that since local patterns are considered similar (*i.e.* filled with mismatches due to noisy object edges), the highlighted region contains visible false negatives.

for the specified area, we can notice that even though the the input image (bottom row) is clearly different from the model (top row), some pixels are still falsely labeled as background since their intra-LBSP and inter-LBSP binary vectors are only slightly different. Here, both of these patterns (shown respectively in 1.b and 1.d) are almost filled with mismatches and are thus similar; this is due to the nonrepresentative reference intensity used for local comparisons (the "central" pixel in 1.b). Directly comparing this same intensity to its new value in the input frame (the "central" pixel in 1.d) would however indicate that the pixel should be considered foreground. Therefore, label assignment should also rely on direct color intensity comparisons in order to reduce the false negative rate of our final method. Also, using this new approach, we capitalize on the under-exploited pixel intensity information already present in the model at a negligible cost. A simplified and concise version of this two-step verification is presented in section 3.2.

Next, let us be reminded that the LBSP descriptor is based on absolute differences and values are considered similar if they fall within a given threshold of each other. In [3], an absolute threshold T_d is used and the general grayscale image description rule is presented as :

$$LBSP(x,y) = \sum_{p=0}^{P-1} d_{T_d}(i_{x,y,p}\, ,\, i_{x,y}) \cdot 2^p \tag{1}$$

with

$$d_{T_d}(i\, ,\, i_c) = \begin{cases} 1 & \text{if } |i - i_c| \leq T_d \\ 0 & \text{otherwise} \end{cases}\quad , \tag{2}$$

where $i_{x,y}$ corresponds to the intensity of the "central" (x, y) pixel (whether from the current image for intra-LBSP or from a reference frame for inter-LBSP), and $i_{x,y,p}$ corresponds to the intensity of the pth neighbor of (x, y) on a predefined pattern (but always in the current image). Replacing T_d by a value that is relative to $i_{x,y}$ could improve the specificity of the descriptor in high illumination variation cases; a similar proposal was already studied for local ternary patterns in [14]. In our case, we would simply need to change the description rule to :

$$LBSP(x,y) = \sum_{p=0}^{P-1} d_{T_r}(i_{x,y,p}\, ,\, i_{x,y}) \cdot 2^p \tag{3}$$

with

$$d_{T_r}(i\, ,\, i_c) = \begin{cases} 1 & \text{if } |i - i_c| \leq T_r \cdot i_c \\ 0 & \text{otherwise} \end{cases}\quad , \tag{4}$$

where T_r would be the new relative threshold value.

3.2. Segmentation and Maintenance Rules

As we mentioned earlier, the ViBe method relies on the collection and maintenance of background samples using a random approach in order to determine if new samples fit its model by counting their nearest neighbors within an intersection threshold. The original authors already proposed numerous improvements related to distance calculations and post-processing operations in [5]; however, we opted to ignore those in order to keep our own method as simple as possible. Instead, we propose a few tweaks to their original technique that globally improve our own results, all the while trying to keep the method's computational complexity as low as possible.

First of all, we noted during preliminary experiments that using an L2 distance, or Euclidean distance, as proposed in [1] to calculate the similarity between two multi-channel samples is not only an expensive operation, but it also results in worse overall performance than a simpler L1 distance, or city-block distance. Hence, we decided to use the L1 distance whenever we needed to directly calculate the similarity between samples, such as for the color intensity comparison improvement stated earlier.

On a similar note, we also determined that since the cost of sampling and comparing is increased because of the LBSP descriptor itself, avoiding unnecessary computations by using a per-channel strategy becomes much more profitable than before. Hence, for an inbound multi-channel frame, we use the simplified[1] approach presented in Algorithm 1 to determine which pixels match the reference model. Note that in this context, $MDesc_{x,y|c}$ and $MInt_{x,y|c}$ respectively return the intra-LBSP descriptor and color intensity of the *c* channel at position *(x,y)*

[1]The concept of multiple samples is ignored for conciseness.

using the reference model, $\oplus$ is the Hamming distance (XOR+popcount) operator, $LBSP_c(x,y)$ calculates the inter-LBSP binary vector at position *(x,y)* of the *c* channel (which implicitly uses $MInt_{x,y|c}$ for its central pixel intensity), T_{desc} and T_{int} are, respectively, the minimum LBSP descriptor and color intensity variation thresholds to consider a pixel foreground, and K is a constant threshold adjustment factor.

Algorithm 1 Simplified per-channel change detection for FG/BG segmentation, with complementary intensity check

```
1: procedure ASSIGNLABELFORPIXEL($p_{x,y}$)
2:     $TotIntDist \leftarrow 0$
3:     $TotDescDist \leftarrow 0$
4:     for $c \leftarrow 1 : TotalNbChannels$ do
5:         $IntDist \leftarrow L1Dist(MInt_{x,y|c}, i_{x,y|c})$
6:         if $IntDist \geq T_{int} \cdot K$
7:             return; $p_{x,y}$ is foreground.
8:         $DescDist \leftarrow MDesc_{x,y|c} \oplus LBSP_c(x,y)$
9:         if $DescDist \geq T_{desc} \cdot K$
10:            return; $p_{x,y}$ is foreground.
11:        $TotIntDist \leftarrow TotIntDist + IntDist$
12:        $TotDescDist \leftarrow TotDescDist + DescDist$
13:    end for
14:    if $TotIntDist \leq T_{int} \wedge TotDescDist \leq T_{desc}$
15:        return; $p_{x,y}$ is background.
16:    else
17:        return; $p_{x,y}$ is foreground.
```

With this detection strategy, when $K < 1$ and in most cases where the actual foreground is very different from the background, multiple operations can be saved if the evaluation of one of the channels immediately returns a result. For example, if we require at least a 9-bit disturbance in LBSP binary vectors ($T_{desc} = 9$) to consider the pixel of an RGB image as foreground while using $K = 1/3$, it will instantly be labeled as foreground if any of its channels returns a change greater or equal to 3 bits ($T_{desc} \cdot K = 3$), no matter what the total variation might be. In worst-case scenarios (*i.e.* when the frame is a perfect match with the model), only $2 \times TotalNbChannels$ extra integer comparisons per pixel are used. While this improvement is primarily focused on processing speed, another benefit is that when K is small enough, we are also able to detect subtle variations present in a single channel (such as a dark blue object occluding a black background), which was not always the case in the previous base implementations. Multiple evaluations of this modification as a stand-alone improvement on our method are presented in section 4.2.

4. Evaluation

We decided to measure the performance of our new method called "LOcal Binary Similarity segmenTER" (LOBSTER) by using the dataset of the 2012 Change Detection Workshop[7]: it consists of 31 real-world sequences spanning six different categories, each with its own challenges, along with standardized evaluation tools. The main advantage of this approach is that we can easily compare our results to numerous other state-of-the-art methods that have already been ranked based on the following metrics:

- True Negatives (TN) : Number of pixels correctly labeled as background
- True Positives (TP) : Number of pixels correctly labeled as foreground
- False Positives (FP) : Number of pixels incorrectly labeled as foreground
- False Negatives (FN) : Number of pixels incorrectly labeled as background
- Recall (Re) : $TP/(TP + FN)$ (larger is better)
- Specificity (Sp) : $TN/(TN + FP)$ (larger is better)
- False Positive Rate (FPR) : $FP/(FP + TN)$ (smaller is better)
- False Negative Rate (FNR) : $FN/(FP + TN)$[2] (smaller is better)
- Percentage of Wrong Classifications (PWC/PBC) : $100 \cdot (FN + FP)/(TP + FN + FP + TN)$ (smaller is better)
- Precision (Pr) : $TP/(TP + FP)$ (larger is better)
- F-Measure $(F1)$: $(2 \cdot Pr \cdot Re)/(Pr + Re)$ (larger is better)

For more information about the performance metrics or the video categories, refer to [7]. As a side note, no pre-processing step is used on any sequence, and only a 9x9 median blur post-processing step is used to reduce blinking pixel noise. Furthermore, we treat thermal frames as grayscale, and our method automatically adjusts the various thresholds in accordance.

In total, using the different improvements discussed earlier, we decided to test four configurations of our method using both relative ("Rel") and absolute ("Abs") LBSP descriptor thresholds, labeled as :

- A : Uses all available improvements[3]
- B : Uses per-channel change detection
- C : Uses change validation via color intensity
- D : "Straightforward" (LBSP integration only)

4.1. Parameters

Like a few other *spatial* and *spatiotemporal-based* methods, we can expect that all our configurations will remain more sensitive to noise and dynamic background elements

[2]The true FNR should be calculated with $FN/(FN + TP)$, but we used it the way it was defined in [7].

[3]The source code of this implementation based on OpenCV is available online at https://bitbucket.org/pierre_luc_st_charles/lobster

than regular *pixel-based* methods. Indeed, since we primarily use a feature descriptor that produces 16-bit codes in a 5x5 area, we would have to carefully adjust our parameters in every scenario in order to obtain truly optimal performance. However, in order to respect the CDNet competition rules, we rely on the following universal parameter set:

- $T_{desc} = 12$ (for RGB) : threshold used to determine if an input pixel matches the model based on the Hamming distance of 16-bit LBSP binary vectors.
- $T_{int} = 90$ (for RGB) : threshold used to determine if an input pixel matches the model based on the L1 distance of 8-bit intensity values.
- $N = 35$: number of LBSP descriptors and color intensities stored as samples for each pixel in the reference model (similar to ViBe's).
- $\#_{min} = 2$: required number of matching samples to label an input pixel as background (similar to ViBe's).
- $\phi = 16$: default time subsampling factor (similar to ViBe's). Note: we force this parameter to one for the first 100 frames of any sequence when bootstrapping.
- $K = 0.5$: threshold adjustment factor applied to T_{desc} and T_{int}.
- $T_r = 0.365$: relative LBSP descriptor threshold.
- $T_d = 30$: absolute LBSP descriptor threshold.

4.2. CDNet Results

First, we present in Figure 2 a scatter graph of the [PWC,$F1$] metrics pairs obtained by processing the entire CDNet dataset using different approaches. For ViBe and ViBe+, we directly evaluated the authors' results provided for [7]. We can observe that for our configurations, both A_{Rel} and A_{Abs} seem to provide, respectively, the best "wrong classifications to F-Measure" ratio and the best overall F-Measure. Also, as expected, the stand-alone use of the per-channel change detection improvement (B) does not seem to have an effect as important as only using the change validation via color intensity improvement (C). However, both of these still manage to drastically improve the results of the straightforward implementation (D), which, by itself, already outperformed the original ViBe method. It is also interesting to point out that replacing the default absolute LBSP threshold by a relative LBSP threshold is not always beneficial with these parameters, as only the A and C configurations seem to really benefit from it.

Next, in Table 2, we present the complete averaged results of our best configuration (A_{Rel}) from Figure 2) along with, in parentheses, each measure variation when compared to the original ViBe method. Globally, we can see that the thermal and cameraJitter categories display the best improvements while the dynamicBackground performance variations seem to stay generally balanced with ViBe's (most likely as a side-effect of the increased sensitivity).

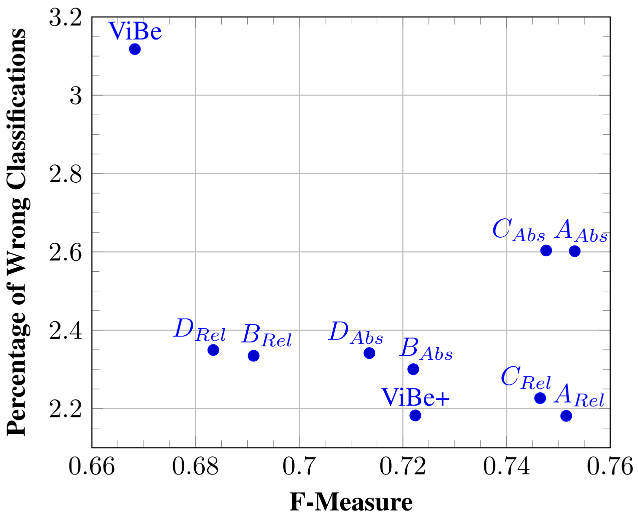

Figure 2. Averaged overall [PWC,F1] metrics pairs obtained on the complete CDNet dataset (where best = bottom right).

Method	Memory (Bytes Per RGB Pixel, Per Sample)	320x240 relative average FPS
LOBSTER (C_{Rel})	72	0.122
LOBSTER (A_{Rel})	72	0.147
LOBSTER (C_{Abs})	72	0.151
LOBSTER (A_{Abs})	72	0.181
ViBe	24	1.000

Table 1. Approximate memory usage and processing speed comparisons for different algorithms when using similar parameters. ViBe speed results were obtained using an optimized home-made implementation that offered results almost identical to CDNet's.

Also, in Table 3, we present the overall averaged results of this same configuration (A_{Rel}) along with its absolute-LBSP counterpart (A_{Abs}), lined up against the other state-of-the-art algorithms that were tested in [7]. Here we can note that our proposed relative-LBSP-threshold-based method outperforms most of them, including the improved ViBe algorithm (ViBe+).

Finally, in Table 1, we also compare the memory usage and speed of our proposed configurations to the original ViBe method. From these results, is it clear that our algorithm appears much slower and bulkier than its counterpart, even with the per-channel change detection improvement. However, we could argue that only comparing it to one of the most lightweight and efficient background subtraction techniques does not do it justice. Indeed, even with an implementation that could still use more optimization, we managed to process the entire CDNet dataset at an average of over 25 frames per second on a rather old dual-core laptop processor (Intel P8600), which could be con-

Category	Recall	Specificity	FPR	FNR	PWC	Precision	F-Measure
baseline	0.8955 ($\Uparrow$9%)	0.9983 ($\approx$0%)	0.0017 ($\Downarrow$17%)	0.0045 ($\Downarrow$40%)	0.5774 ($\Downarrow$35%)	0.9558 ($\Uparrow$3%)	0.9242 ($\Uparrow$6%)
cameraJitt	0.6742 ($\Downarrow$5%)	0.9921 ($\Uparrow$2%)	0.0079 ($\Downarrow$74%)	0.0141 ($\Uparrow$22%)	2.0930 ($\Downarrow$48%)	0.8368 ($\Uparrow$58%)	0.7423 ($\Uparrow$24%)
dynamicBg	0.7670 ($\Uparrow$6%)	0.9820 ($\Downarrow$1%)	0.0180 ($\Uparrow$73%)	0.0023 ($\Downarrow$15%)	1.9984 ($\Uparrow$56%)	0.5923 ($\Uparrow$11%)	0.5679 ($\approx$0%)
intermittObj	0.5589 ($\Uparrow$9%)	0.9752 ($\Uparrow$2%)	0.0248 ($\Downarrow$48%)	0.0428 ($\approx$0%)	5.8427 ($\Downarrow$25%)	0.7102 ($\Uparrow$9%)	0.5770 ($\Uparrow$14%)
shadow	0.8784 ($\Uparrow$12%)	0.9930 ($\approx$0%)	0.0070 ($\Downarrow$14%)	0.0052 ($\Downarrow$44%)	1.1549 ($\Downarrow$30%)	0.8765 ($\Uparrow$5%)	0.8728 ($\Uparrow$9%)
thermal	0.8135 ($\Uparrow$49%)	0.9934 ($\approx$0%)	0.0066 ($\Uparrow$72%)	0.0093 ($\Downarrow$71%)	1.4210 ($\Downarrow$55%)	0.8572 ($\Downarrow$8%)	0.8248 ($\Uparrow$24%)
overall	0.7646 ($\Uparrow$12%)	0.9890 ($\Uparrow$1%)	0.0110 ($\Downarrow$35%)	0.0130 ($\Downarrow$26%)	2.1812 ($\Downarrow$30%)	0.8048 ($\Uparrow$9%)	0.7515 ($\Uparrow$13%)

Table 2. Complete results obtained with the proposed fully improved, relative-LBSP-based method (A_{Rel}) on the CDNet dataset, with, in parentheses, the variation when compared to the original CDNet ViBe results.

Method	Recall	Specificity	FPR	FNR	PWC	Precision	F-Measure	Ranking
PBAS[9]	0.78	0.990	0.010	0.009	1.77	0.82	0.75	3.43
LOBSTER (A_{Rel})	0.76	0.989	0.011	0.013	2.18	0.80	0.75	5.00
ViBe+[5]	0.69	0.993	0.007	0.017	2.18	0.83	0.72	5.43
PSP-MRF[21]	0.80	0.983	0.017	0.009	2.39	0.75	0.74	6.00
SC-SOBS[16]	0.80	0.983	0.017	0.009	2.41	0.73	0.73	6.86
Chebyshev Prob.[18]	0.71	0.989	0.011	0.015	2.39	0.79	0.70	7.00
LOBSTER (A_{Abs})	0.80	0.982	0.019	0.010	2.60	0.77	0.75	7.71
SOBS[15]	0.79	0.982	0.018	0.009	2.56	0.72	0.72	9.00
KDE Nonaka *et al.* [19]	0.65	0.993	0.007	0.025	2.89	0.77	0.64	9.71
GMM KaewTraKulPong[12]	0.51	0.995	0.005	0.029	3.11	0.82	0.59	10.57
KDE Yoshinaga *et al.* [27]	0.66	0.991	0.009	0.024	3.00	0.73	0.64	10.71
GMM Stauffer-Grimson[23]	0.71	0.986	0.014	0.020	3.10	0.70	0.66	11.29
ViBe[1]	0.68	0.983	0.017	0.018	3.12	0.74	0.67	11.57
GMM Zivkovic[30]	0.70	0.984	0.016	0.019	3.15	0.71	0.66	12.29
KDE Elgammal[6]	0.74	0.976	0.024	0.014	3.46	0.68	0.67	13.43
Bayesian Background[20]	0.60	0.983	0.017	0.020	3.39	0.74	0.63	13.43
Local-Self Similarity[11]	0.94	0.851	0.149	0.002	14.30	0.41	0.50	14.57
GMM RECTGAUSS-Tex[25]	0.52	0.986	0.014	0.027	3.68	0.72	0.52	14.71
Mahalanobis distance[2]	0.76	0.960	0.040	0.011	4.66	0.60	0.63	15.14
Euclidean distance[2]	0.70	0.969	0.031	0.017	4.35	0.62	0.61	16.00

Table 3. New complete overall results and updated ranks after manually adding our two best method configurations to the 2012 CDNet Workshop rankings.

sidered extremely fast when compared to other *spatial* and *spatiotemporal-based* methods.

5. Conclusion

In this paper, we presented a novel and highly efficient *spatiotemporal-based* background subtraction algorithm named LOBSTER that is based on a modified ViBe model and improved LBSP features that outperforms much more complex state-of-the-art algorithms, including its original counterpart's improved version (ViBe+). In doing so, we showed that simply incorporating an LBSP feature component in a regular *pixel-based* method could result in dramatic performance increase at an acceptable cost. For future work, the integration of LBSP features and the proposed modifications in other similar methods (such as PBAS) could prove even more successful. Also, optimizing the different components of our algorithm using SIMD instructions or even completely reimplementing them on GPU could significantly close the speed gap between our method and other lightweight *pixel-based* methods.

6. Acknowledgements

This work was supported by NSERC discovery grant No. 311869-2010.

References

[1] O. Barnich and M. Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *Image Processing, IEEE Transactions on*, 20(6):1709–1724, 2011. 1, 2, 3, 6

[2] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Comparative study of background subtraction algorithms. *Journal of Electronic Imaging*, 19, 07 2010. 6

[3] G.-A. Bilodeau, J.-P. Jodoin, and N. Saunier. Change detection in feature space using local binary similarity patterns. In *Computer and Robot Vision (CRV), 2013 International Conference on*, pages 106–112, 2013. 1, 2, 3

[4] Y.-T. Chen, C.-S. Chen, C.-R. Huang, and Y.-P. Hung. Efficient hierarchical method for background subtraction. *Pattern Recognition*, 40(10):2706 – 2715, 2007. 2

[5] M. V. Droogenbroeck and O. Paquot. Background subtraction: Experiments and improvements for vibe. In *CVPR Workshops 2012*, pages 32–37. IEEE, 2012. 1, 2, 3, 6

[6] A. M. Elgammal, D. Harwood, and L. S. Davis. Non-parametric model for background subtraction. In *Proceedings of the 6th European Conference on Computer Vision-Part II*, ECCV '00, pages 751–767, London, UK, UK, 2000. Springer-Verlag. 1, 6

[7] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. Changedetection.net: A new change detection benchmark dataset. In *CVPR Workshops 2012*, pages 1–8, 2012. 1, 4, 5

[8] M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):657–662, 2006. 2

[9] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *CVPR Workshops 2012*, pages 38–43, 2012. 2, 6

[10] S.-S. Huang, L.-C. Fu, and P.-Y. Hsiao. Region-level motion-based background modeling and subtraction using mrfs. *Image Processing, IEEE Transactions on*, 16(5):1446–1456, 2007. 2

[11] J.-P. Jodoin, G.-A. Bilodeau, and N. Saunier. Background subtraction based on local shape. *CoRR*, abs/1204.6326, 2012. 6

[12] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In P. Remagnino, G. Jones, N. Paragios, and C. Regazzoni, editors, *Video-Based Surveillance Systems*, pages 135–144. Springer US, 2002. 6

[13] W. Kim and C. Kim. Background subtraction for dynamic texture scenes using fuzzy color histograms. *Signal Processing Letters, IEEE*, 19(3):127–130, 2012. 2

[14] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikainen, and S. Li. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In *CVPR 2010*, pages 1301–1306, 2010. 3

[15] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *Trans. Img. Proc.*, 17(7):1168–1177, July 2008. 6

[16] L. Maddalena and A. Petrosino. The sobs algorithm: What are the limits? In *CVPR Workshops 2012*, pages 21–26, 2012. 6

[17] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *CVPR 2004*, volume 2, pages II–302–II–309 Vol.2, 2004. 2

[18] A. Morde, X. Ma, and S. Guler. Learning a background model for change detection. In *CVPR Workshops 2012*, pages 15–20, 2012. 6

[19] Y. Nonaka, A. Shimada, H. Nagahara, and R. Taniguchi. Evaluation report of integrated background modeling based on spatio-temporal features. In *CVPR Workshops 2012*, pages 9–14, 2012. 6

[20] F. Porikli and O. Tuzel. Bayesian background modeling for foreground detection. In *Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, VSSN '05, pages 55–58, New York, NY, USA, 2005. ACM. 6

[21] A. Schick, M. Bauml, and R. Stiefelhagen. Improving foreground segmentations with probabilistic superpixel markov random fields. In *CVPR Workshops 2012*, pages 27–31, 2012. 6

[22] A. Shimada, H. Nagahara, and R. Taniguchi. Background modeling based on bidirectional analysis. In *CVPR 2013*, pages 1979–1986, 6 2013. 2

[23] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR 1999*, volume 2, pages –252 Vol. 2, 1999. 1, 6

[24] T. Tanaka, A. Shimada, R.-i. Taniguchi, T. Yamashita, and D. Arita. Towards robust object detection: integrated background modeling based on spatio-temporal features. In *Proceedings of the 9th Asian conference on Computer Vision - Volume Part I*, ACCV'09, pages 201–212, Berlin, Heidelberg, 2010. Springer-Verlag. 2

[25] P. D. Z. Varcheie, M. Sills-Lavoie, and G.-A. Bilodeau. A multiscale region-based motion detection and background subtraction algorithm. *Sensors*, 10(2):1041–1061, 2010. 2, 6

[26] M. Wu and X. Peng. Spatio-temporal context for codebook-based dynamic background subtraction. *{AEU} - International Journal of Electronics and Communications*, 64(8):739 – 747, 2010. 2

[27] S. Yoshinaga, A. Shimada, H. Nagahara, and R. Taniguchi. Background model based on intensity change similarity among pixels. In *Frontiers of Computer Vision, (FCV), 2013 19th Korea-Japan Joint Workshop on*, pages 276–280, 2013. 6

[28] S. Yoshinaga, A. Shimada, H. Nagahara, and R.-i. Taniguchi. Object detection using local difference patterns. In *Proceedings of the 10th Asian conference on Computer vision - Volume Part IV*, ACCV'10, pages 216–227, Berlin, Heidelberg, 2011. Springer-Verlag. 2

[29] S. Zhang, H. Yao, and S. Liu. Dynamic background modeling and subtraction using spatio-temporal local binary patterns. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1556–1559, 2008. 2

[30] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31 Vol.2, 2004. 2, 6