Flexible Background Subtraction With Self-Balanced Local Sensitivity

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

Robert Bergevin
LVSN - REPARTI
Université Laval
Québec City, QC, Canada
robert.bergevin@gel.ulaval.ca

Abstract

Most background subtraction approaches offer decent results in baseline scenarios, but adaptive and flexible solutions are still uncommon as many require scenario-specific parameter tuning to achieve optimal performance. In this paper, we introduce a new strategy to tackle this problem that focuses on balancing the inner workings of a non-parametric model based on pixel-level feedback loops. Pixels are modeled using a spatiotemporal feature descriptor for increased sensitivity. Using the video sequences and ground truth annotations of the 2012 and 2014 CVPR Change Detection Workshops, we demonstrate that our approach outperforms all previously ranked methods in the original dataset while achieving good results in the most recent one.

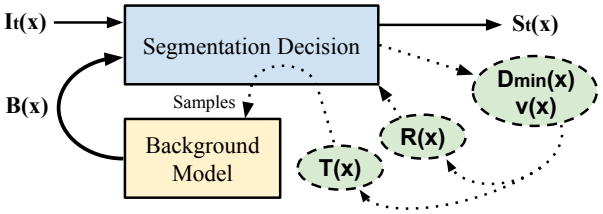

Figure 1: Basic overview of SuBSENSE's principal components; dotted lines indicate feedback mechanisms. In this context, $I_t(x)$ carries the LBSP/RGB representation of x obtained from the t-th frame of the analyzed sequence, $B(x)$ contains N recently observed background samples, $S_t(x)$ is the segmentation output value, $R(x)$ controls the internal segmentation decision thresholds, $T(x)$ controls the background update thresholds, and finally, both $D_{min}(x)$ and $v(x)$ dynamically control the previous variables by monitoring background dynamics.

1. Introduction

Background subtraction algorithms are often the first step of more in-depth video analysis procedures. Most approaches require their own set of finely tuned, scenario-specific parameters in order to achieve optimal results. However, finding the best parameters for a given application is often disregarded as it is not always trivial: it requires good knowledge of the data and of the algorithm itself. Thus, this crucial component can easily become a bottleneck when used in complex systems. In this paper, we present a new background subtraction approach that relies on the automatic adjustment of maintenance, update and labeling rules surrounding a non-parametric model. Our goal is to provide optimal segmentation results across multiple types of scenarios simultaneously. Our contributions are highlighted in the two logical steps of our algorithm (see Fig. 1 for a simplified overview): first, pixel-level change detection is done by means of spatiotemporal analyses based on color and Local Binary Similarity Patterns (LBSP) [3, 16]. This allows increased sensitivity for change detection. Then, the essence of our method's flexibility resides in its automatic adjustments of local sensitivity: decision thresholds and state variables are continuously affected by pixel-level feedback loops originating from previous analyses. These include typical noise compensation schemes, but also include blinking pixel and ghost detection policies.

Coined "SuBSENSE" (Self-Balanced SENsitivity SEgmenter), this new method allows us to identify and isolate areas where segmentation is more difficult. Those areas can then be treated differently from other frame regions, which might only exhibit common behavior and require standard treatment (see Fig. 2 for an example). A complete evaluation using the 2012 www.ChangeDetection.net (CDNet) dataset [6] shows that we are able to achieve excellent overall performance in difficult scenarios, outranking all previous approaches, even for baseline sequences. This means that flexibility is not gained at the expense of performance on simpler scenarios. As for the 2014 update of this same dataset, comparisons with the methods that were already

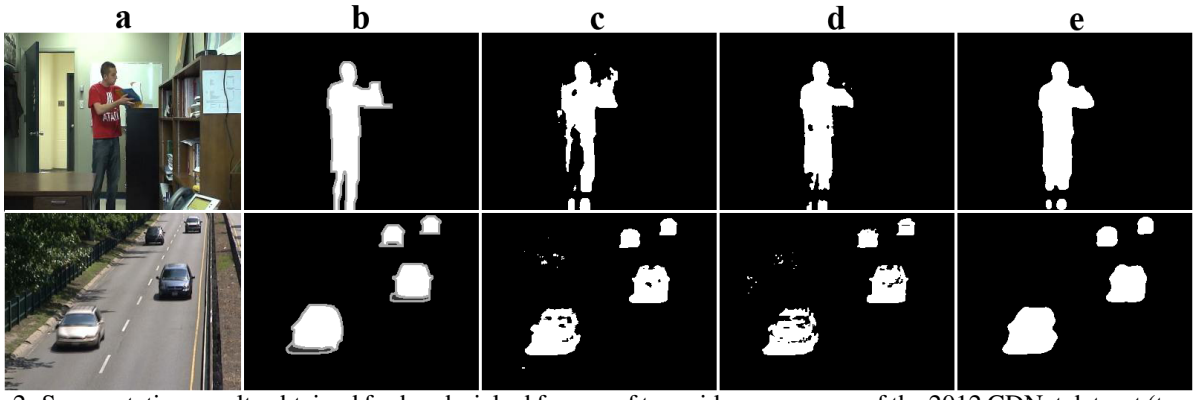

Figure 2: Segmentation results obtained for hand-picked frames of two video sequences of the 2012 CDNet dataset (top row is "office" at frame 966, bottom row is "highway" at frame 1091) using two well-known methods, Stauffer and Grimson's GMM [17] and ViBe [1], as well as our proposed method. Column (a) shows the input frames, (b) the ground truth segmentation map, (c) GMM's results, (d) ViBe's results and (e) our own results. Since GMM and ViBe cannot automatically adapt to all possible segmentation scenarios, a middle ground must be established to allow good overall flexibility.

ranked (as of 2014/05/05) show that our approach is also generally a better alternative, even in the most difficult conditions. Additionally, the processing speed of our algorithm is still widely acceptable for real-time applications.

2. Background and Motivation

There have been countless different approaches proposed over the years for background subtraction via change detection; our method is inspired by some of the ideas put forward by contenders of the original CDNet workshop. First up are non-parametric approaches based on sample consensus: ViBe [1], ViBe+ [4] and PBAS [8] are all good examples of simple methods that only rely on the pixel-level sampling of colors to create their background model. Although far less complex than methods based on probabilistic models, these performed quite well on the 2012 CDNet dataset, as reported in [6]. Therefore, we decided to also use a sample consensus model for our own method. Besides, this kind of model is very well suited for the description of pixels via complex features, as no aggregation is required.

PBAS is also the main inspiration of our feedback scheme: unlike regular methods which use a global strategy for model maintenance and internal labeling decisions, Hofmann *et al*. proposed in [8] to monitor background dynamics at the pixel level. This allowed them to identify and treat unstable regions (*i.e*. regions that the model cannot properly adapt to) by readjusting distance thresholds and adaptation rates locally using feedback mechanisms. Their monitoring technique is however inefficient against regions with intermittent dynamic variations (*e.g*. tree branches on a windy day), thus why we decided to rework their approach in our own novel method.

Lastly, Local Binary Similarity Pattern (LBSP)-based features were demonstrated by Bilodeau *et al*. in [3] to be more effective than color intensities at detecting relevant pixel-level changes in simple background subtraction scenarios. These are a variation of Local Binary Patterns, which were first used in [7] to improve robustness against illumination variations and reduce false classifications due to camouflaged foreground objects. The particularity of LBSP features is that they can also be computed across multiple frames, improving temporal consistency. For this reason, we incorporated them in our model to improve the spatiotemporal sensitivity of our approach.

3. Self-Balanced Sensitivity Segmenter

Essentially, we use a sample-based, non-parametric statistical model that portrays the background at individual pixel locations (noted $B(x)$) using a set of $N = 50$ past representations (or "samples"). When at least $\#_{min} = 2$ samples intersect with the representation of x at time t (noted $I_t(x)$), the pixel is labeled as background in the raw segmentation map ($S_t(x) = 0$); otherwise, it is automatically considered foreground ($S_t(x) = 1$). The way these background "samples" are updated is similar to what is done in [1, 8]: they are randomly replaced by local values after the segmentation step, but only when $S_t(x) = 0$.

In their most basic form, the samples in $B(x)$ could only contain pixel colors, and the comparison scheme used to compute intersections between $B(x)$ and $I_t(x)$ could simply be based on L1 or L2 distance, using a specific maximum threshold. However, we directly incorporate binary strings obtained from LBSP features to our model, resulting in samples that carry both local color intensity and spatiotemporal neighborhood similarity information. More specifically, for a regular RGB image, our samples consist of 8-bit RGB intensities paired with 16-bit LBSP binary strings (for each channel). Accordingly, comparisons are

done on colors using L1 distance and on binary strings using Hamming distance. Although the maximum distance thresholds (noted R_{color} and R_{lbsp}) used for these two operations differ in nature, both are obtained from the same abstract threshold variable, noted $R(x)$:

$$R_{color} = R(x) \cdot R^0_{color}, \tag{1}$$

$$R_{lbsp} = 2^{R(x)} + R^0_{lbsp}, \tag{2}$$

where $R(x) \geq 1$, and R^0_{color} and R^0_{lbsp} are, respectively, the minimal color and LBSP distance thresholds (30 and 3 in our case). We define $R(x)$ as a pixel-level variable that controls the maximum difference allowed between two representations (or samples) before they are considered dissimilar. This variable is dynamically adjusted using the moving average of recent minimal distances between $B(x)$ and $I_t(x)$, noted $D_{min}(x)$. The following equation reflects how $D_{min}(x)$ governs $R(x)$ and how $R(x)$ is updated:

$$R(x) = \begin{cases} R(x) + v(x) & \text{if } R(x) < (1+D_{min}(x)\cdot 2)^2 \\ R(x) - \frac{1}{v(x)} & \text{otherwise} \end{cases}, \tag{3}$$

where $v(x)$ is a strictly-positive factor (described further down). We determined through early experiments that modifying $R(x)$ using a relative value (as suggested in [8]) generally results in a slower feedback response time than using an absolute value (based on $v(x)$ in our case). Also, we established the link between $D_{min}(x)$ and $R(x)$ using an exponential relation: this allows the use of much larger $R(x)$ threshold values when $D_{min}(x)$ becomes more significant. Eq. 4 shows how $D_{min}(x)$ itself is updated using the current normalized minimal distances between samples in $B(x)$ and $I_t(x)$ (noted $d_t(x)$), where α is a predetermined constant learning rate. Note that $D_{min}(x)$ is always bound to the $[0, 1]$ interval.

$$D_{min}(x) = D_{min}(x)\cdot(1-\alpha) + d_t(x)\cdot\alpha \tag{4}$$

The concept behind $R(x)$'s relation with $D_{min}(x)$ is simple: an area with continuous change (caused, for example, by rippling water or falling snow) will possess a higher $D_{min}(x)$ than an area with very little change. Therefore, its local $R(x)$ value will be greater (as dictated through Eq. 3), and intersections of background samples with $I_t(x)$ will also be more likely.

Unlike PBAS's approach ([8]), our main pixel-level monitoring variable ($D_{min}(x)$) is updated using Eq. 4 for every new observation, no matter what the actual pixel classification is. This results in faster threshold responses to intermittent dynamic elements (such as swaying tree branches due to wind bursts), which drastically reduce the amount of false detections in unstable areas. However, it also leads to rapidly increasing $D_{min}(x)$ values (and, logically, $R(x)$) in regions where foreground objects frequently occlude the background. This is due to the fact that the model can only slowly incorporate new observed samples in $B(x)$, and continuous disparities between $B(x)$ and $I(x)$ lead to $D_{min}(x)$ buildups. Such behavior may become detrimental over time, as true foreground objects would become harder and harder to detect.

To counter this problem, we propose to dynamically readjust the $v(x)$ value seen in Eq. 3 to control the variations of $R(x)$ based on the region's nature. Ideally, static regions should be characterized by small $v(x)$ values ($0 < v(x) < 1$), thus making sure $R(x)$ can barely increase but may decrease rapidly. On the other hand, $v(x)$ should be much greater in dynamic regions ($v(x) > 1$), leading to the opposite phenomenon. To properly discern and isolate static and dynamic regions over time, we rely on the analysis of blinking pixels. These are characteristic of dynamic regions, as segmentation noise is often intermittent and irregular in such areas. Therefore, an XOR operation on subsequent segmentation frames (S_t and S_{t-1}) can easily reveal blinking pixels. We use $v(x)$ variables as accumulators in this context: each time a blinking pixel is seen at x, we increase the value of $v(x)$ and we slowly decrease it otherwise.

As stated earlier, the samples contained in $B(x)$ can be randomly replaced by local representations if the current intersection of $B(x)$ and $I_t(x)$ resulted in $\#_{min}$ or more matches. We follow part of the logic that was suggested in [1, 4, 8] in this matter: for every background classification, a single, randomly-picked sample of $B(x)$ (as well as one of its direct neighbors') can be replaced with the actual observation of x based on probability $p = 1/T(x)$ (where $T(x)$ is the "update rate"). In our case, $T(x)$ also depends on our two background dynamics monitoring variables, $D_{min}(x)$ and $v(x)$, and on the latest segmentation results, $S_t(x)$. More specifically, we use:

$$T(x) = \begin{cases} T(x) + \frac{1}{v(x)\cdot D_{min}(x)} & \text{if } S_t(x) = 1 \\ T(x) - \frac{v(x)}{D_{min}(x)} & \text{if } S_t(x) = 0 \end{cases}, \tag{5}$$

where $T(x)$ is limited to the $[T_{lower}, T_{upper}]$ interval (by default, $[2, 256]$). In short, this relation dictates that background regions with very little variation should see rapid update rate increases (*i.e.* sudden drops in update probability) when they are classified as foreground, whereas unstable areas should experience the opposite (*i.e.* slower increases in $T(x)$). The reasoning behind raising the update rate (and reducing the update probability) for foreground regions is that the model should try to keep older local representations active for as long as possible in order to allow the continuous foreground segmentation of intermittently moving objects of interest.

On a related note, the detection of "ghosts" is very challenging in background subtraction, especially at the pixel level: these typically appear when part of the background

Categories	*Recall*	*Specificity*	*FPR*	*FNR*	*PWC*	*Precision*	*F-Measure*
baseline	0.9622	0.9976	0.0024	0.0378	0.3821	0.9346	0.9480
cameraJitter	0.7495	0.9908	0.0092	0.2505	1.8282	0.8116	0.7694
dynamicBackground	0.7872	0.9993	0.0007	0.2128	0.3837	0.8768	0.8138
intermObjectMotion	0.6679	0.9919	0.0081	0.3321	3.7722	0.7975	0.6523
shadow	0.9529	0.9910	0.0090	0.0471	1.0668	0.8370	0.8890
thermal	0.8379	0.9889	0.0111	0.1621	1.6145	0.8116	0.8184
overall (2012)	0.8263	0.9933	0.0067	0.1737	1.5079	0.8449	0.8152
badWeather	0.8100	0.9989	0.0011	0.1900	0.4671	0.9051	0.8528
lowFramerate	0.8399	0.9944	0.0056	0.1601	0.9644	0.6122	0.6437
nightVideos	0.6262	0.9779	0.0221	0.3738	3.7145	0.5168	0.5390
PTZ	0.8316	0.9418	0.0582	0.1684	5.9293	0.2666	0.3185
turbulence	0.8118	0.9995	0.0005	0.1882	0.1348	0.8398	0.8197
overall (2012+2014)	0.8070	0.9884	0.0116	0.1930	1.8416	0.7463	0.7331

Table 1: Results obtained for SuBSENSE using the 2012 and 2014 versions of the CDNet dataset and evaluation tools.

model portrays an older, but irrelevant version of the observed scene, causing the "true" background to be misclassified as foreground. This usually happens when static background objects are removed from the scene (e.g. a car leaving a parking lot). In our case, for a given x, if $T(x)$ is abruptly raised to T_{upper} (due to $D_{min}(x) \approx 0$), it could take a while before enough samples are added back to $B(x)$ to restart classifying x as background. To solve this problem, we determined that when regions are classified as foreground over long periods of time (over 300 frames) and when local variations between consecutive frames (I_t and I_{t-1}) are negligible, increasing the update probability by temporarily setting $T(x)$ to a lower value (*i.e.* 4) usually helps. Once the region no longer needs updates or no longer matches the "ghost" profile, it regains its regular, dynamic $T(x)$ value.

Segmentation results in sequences where the camera is not completely static are also improved by monitoring the overall observation disparities between consecutive frames in order to dynamically readjust learning rates. Although this solution is not perfect, it allows us to detect sudden and continuous camera movements as long as the observed scene presents enough high-level changes. More specifically, we use moving averages of downsampled input frames (1/8 of the original frame size) to detect approximately when large portions of the analysis region suffer from drastic changes, in both pixel-level intensities and textures. When such an event is detected, the T_{lower} and T_{upper} variables are scaled down based on the severity of this event, thus instantly increasing update probabilities frame-wide.

4. Evaluation

In order to evaluate how our solution performs against various types of background subtraction problems, we first relied on the video sequence dataset proposed in the 2012 CVPR Workshop on Change Detection [6] that includes 31 video sequences spanning six categories: baseline, camera jitter, dynamic background, intermittent object motion, shadow and thermal. We also used the 2014 version of this same dataset, where 22 videos were added in five new categories: bad weather, low framerate, night videos, point-tilt-zoom (PTZ) and turbulence. As required, we used a unique parameter set for all videos to determine the true flexibility of our method. The post-processing operations we used are only based on median blur and morphological operations. Note that the latest C++ implementation of SuBSENSE is available online[1]. The metrics used in the tables below are described in [6].

First, we present in Table 1 the averaged metrics across all 2012 and 2014 categories obtained using our proposed method as well as the averaged overall results. We can observe that our results for the six original categories are much better than those of the newer categories, indicating that the videos added in the 2014 dataset present much more challenging background subtraction problems. In our case, the PTZ category seems to be the worse: since we do not rely on accurate region-level motion estimation, our approach does not handle non-static cameras very well, causing continuous false classifications over the entire analyzed region of interest. Adding a video stabilization step as preprocessing or an optical flow analysis component to our method could improve performance in this case. Segmentation results obtained for "night videos" also seem problematic: since cars are the main focus in this category, their headlights in dark regions often cause important illumination changes that result in false positive classifications. Despite these two drawbacks, all other categories show F-Measures above 60%, which indicates that our approach offers generally good per-

[1] https://bitbucket.org/pierre_luc_st_charles/subsense

Method	AvgRank	FMeasure
SuBSENSE (Proposed)	**2.16**	**0.8152**
PBAS[8]	4.00	0.7532
ViBe+[4]	5.33	0.7224
PSP-MRF[15]	5.50	0.7372
SC-SOBS[11]	6.50	0.7283
Chebyshev Prob.[12]	7.16	0.7001
SOBS[10]	8.83	0.7155
KNN [21]	8.83	0.6785
KDE Nonaka *et al.* [13]	9.50	0.6418
KDE Elgammal *et al.* [5]	9.66	0.6719
ViBe[1]	10.50	0.6683
KDE Yoshinaga *et al.* [19]	11.16	0.6437
Bayesian Background[14]	11.66	0.6272
GMM Stauffer-Grimson[17]	12.00	0.6624
GMM Zivkovic[20]	14.00	0.6596
GMM RECTGAUSS-T.[18]	14.33	0.5221
Local-Self Similarity[9]	14.66	0.5016
Mahalanobis distance[2]	15.50	0.6259
Euclidean distance[2]	17.33	0.6111

Table 2: Updated average ranks (obtained via the MethodsRanker script) and overall F-Measures of the original methods tested on the 2012 CDNet dataset.

Method	AvgRank	FMeasure
FTSG	**2.14**	0.7283
SuBSENSE (Proposed)	2.43	**0.7331**
CwisarDH	4.57	0.6812
Spectral-360	4.86	0.6732
Bing Wang's	6.14	0.6577
KNN [21]	7.43	0.5937
SC-SOBS	7.57	0.5961
Mahalanobis distance[2]	8.29	0.2267
CP3-online	8.43	0.5805
GMM Stauffer-Grimson[17]	8.43	0.5707
KDE Elgammal *et al.* [5]	9.71	0.5688
GMM Zivkovic[20]	10.71	0.5566
MST BG Model	12.00	0.5141
Euclidean distance[2]	12.29	0.5161

Table 3: Average ranks and overall F-Measures of all available methods on the 2014 CDNet dataset as of 2014/05/05; for more information about each method, refer to www.ChangeDetection.net.

formance.

Then, in Tables 2 and 3, we present how our method ranks among other CDNet contenders (based on CDNet's MethodsRanker script for 2012 results and on website rankings for 2014 results) along with their respective overall F-Measure metrics (which were considered good indicators of overall performance in [6]). Although we did not reach the first overall rank in the 2014 dataset, we can denote that our F-Measure score surpasses that of all other state-of-the-art methods. This is due to our algorithm's excellent flexibility in most scenarios, which allows the treatment of different frame regions using the right level of change detection sensitivity. Also, the full results (which are not shown here but available on the CDNet website) indicate that our method obtains the best F-Measure score in seven out of eleven categories.

Besides, even though processing speed was not the main focus of our work, the configuration we used managed to clock an average of over 30 frames per second for the complete 2014 dataset (53 sequences), which is well above the expected average for other methods. For reference, we only used C++ code on a 3rd generation Intel Core i5 processor with no architecture-specific instructions or other low-level optimizations.

5. Conclusion

We presented an adaptive and truly flexible background subtraction algorithm based on pixel-level change detection using comparisons of colors and LBSP features. The feedback-driven, self-balancing properties of this new approach allowed us to reach a new height in terms of global performance on the 2012 CDNet dataset, but our experiments also showed that we outperform all other already-ranked methods in the 2014 version of this same dataset in terms of F-Measure scores. We believe our results could still be drastically improved in PTZ scenarios by adding an accurate region-level analysis component based on motion estimation, or in general by using a Markov Random Field-based post-processing component (such as [15]).

6. Acknowledgements

This work was supported by FRQ-NT team grant No. 2014-PR-172083 and by REPARTI (Regroupement pour l'étude des environnements partagés intelligents répartis) FRQ-NT strategic cluster.

References

[1] O. Barnich and M. Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *Image Processing, 2011 IEEE Transactions on*, 20(6):1709–1724, 2011. 2, 3, 5

[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. 5

[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

[4] M. V. Droogenbroeck and O. Paquot. Background subtraction: Experiments and improvements for vibe. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 32–37. IEEE, 2012. 2, 3, 5

[5] 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 (ECCV)*, pages 751–767, 2000. 5

[6] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. Changedetection.net: A new change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–8, 2012. 1, 2, 4, 5

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

[8] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 38–43, 2012. 2, 3, 5

[9] J.-P. Jodoin, G.-A. Bilodeau, and N. Saunier. Background subtraction based on local shape. *CoRR*, abs/1204.6326, 2012. 5

[10] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168–1177, July 2008. 5

[11] L. Maddalena and A. Petrosino. The sobs algorithm: What are the limits? In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 21–26, 2012. 5

[12] A. Morde, X. Ma, and S. Guler. Learning a background model for change detection. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 15–20, 2012. 5

[13] Y. Nonaka, A. Shimada, H. Nagahara, and R. Taniguchi. Evaluation report of integrated background modeling based on spatio-temporal features. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 9–14, 2012. 5

[14] 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. 5

[15] A. Schick, M. Bauml, and R. Stiefelhagen. Improving foreground segmentations with probabilistic superpixel markov random fields. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 27–31, 2012. 5

[16] P.-L. St-Charles and G.-A. Bilodeau. Improving background subtraction using local binary similarity patterns. In *Applications of Computer Vision (WACV), 2014 IEEE Computer Society Winter Conference on*, 2014. 1

[17] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition (CVPR), 1999 IEEE Computer Society Conference on*, volume 2, pages –252 Vol. 2, 1999. 2, 5

[18] 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. 5

[19] 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. 5

[20] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition (ICPR), 2004 International Conference on*, volume 2, pages 28–31 Vol.2, 2004. 5

[21] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, 2006. 5