

REAL-TIME SEMANTIC BACKGROUND SUBTRACTION

Anthony Cioppa Marc Van Droogenbroeck Marc Braham

Department of Electrical Engineering and Computer Science (Montefiore Institute)
University of Liège, Belgium
{Anthony.Cioppa, M.VanDroogenbroeck, M.Braham}@uliege.be

ABSTRACT

Semantic background subtraction (SBS) has been shown to improve the performance of most background subtraction algorithms by combining them with semantic information, derived from a semantic segmentation network. However, SBS requires high-quality semantic segmentation masks for all frames, which are slow to compute. In addition, most state-of-the-art background subtraction algorithms are not real-time, which makes them unsuitable for real-world applications. In this paper, we present a novel background subtraction algorithm called Real-Time Semantic Background Subtraction (denoted RT-SBS) which extends SBS for real-time constrained applications while keeping similar performances. RT-SBS effectively combines a real-time background subtraction algorithm with high-quality semantic information which can be provided at a slower pace, independently for each pixel. We show that RT-SBS coupled with ViBe sets a new state of the art for real-time background subtraction algorithms and even competes with the non real-time state-of-the-art ones. Note that we provide python CPU and GPU implementations of RT-SBS at <https://github.com/cioppaanthony/rt-sbs>.

Index Terms— background subtraction, semantic segmentation, change detection, real-time processing

1. INTRODUCTION

Background subtraction (BGS) algorithms aim at detecting pixels belonging to moving objects in video sequences [1]. Generally, a BGS algorithm is composed of three elements: an adaptive background model, a similarity criterion to compare a pixel of a frame with the model, and an update strategy for the background model. The BGS algorithm then classifies each pixel of the video into one of the following two classes: foreground (FG) for moving objects, or background (BG).

While many progresses in background subtraction have been achieved since the seminal algorithms GMM [2] and KDE [3], partly due to the availability of pixel-wise annotated datasets such as BMC [4], CDNet 2014 [5] or LASIESTA [6], modern algorithms such as ViBe [7], PAWCS [8], or IUTIS-5 [9] remain sensitive to dynamic backgrounds, illumination

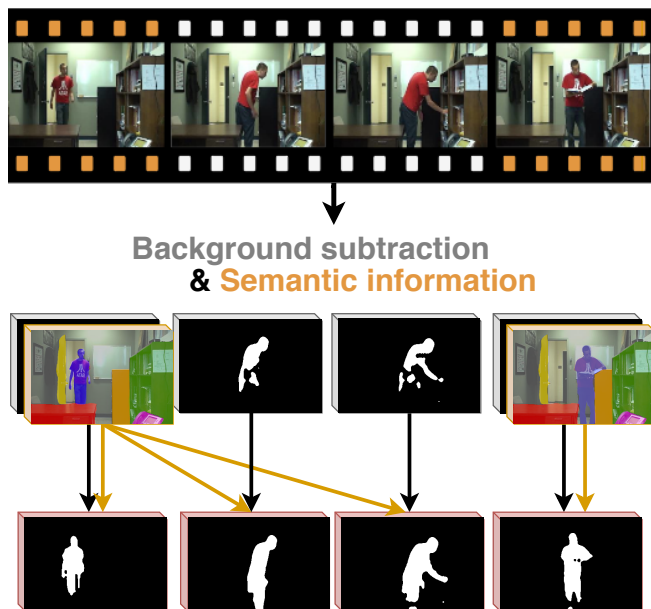


Fig. 1. Our novel background subtraction algorithm, called **RT-SBS**, combines a BGS algorithm with **semantic information** in real time. Semantic information is slow to compute and is only available for some frames, but RT-SBS reuses previous semantic information when appropriate.

changes, shadows, *etc.* Furthermore, most state-of-the-art algorithms are unusable in practice since they are not real-time as stated in [10]. More recently, deep learning based algorithms emerged with the work of Braham *et al.* [11] which opened the path for novel algorithms [12, 13] thanks to the increased power of computers.

In this paper, we focus on a particular BGS algorithm developed by Braham *et al.* [14], called semantic background subtraction (SBS). This algorithm combines the result of a network such as PSPNet [15], used to provide semantic information about objects of interest in the scene, with a BGS algorithm in order to improve the performance of the latter. The objective of semantic segmentation consists in labeling each pixel of an image with a class corresponding to the object that the pixel belongs to. SBS uses semantic information derived

from semantic segmentation to infer whether or not a pixel belongs to a potentially moving object. It has been shown to improve the performance of most unsupervised and, more recently, even supervised BGS algorithms [16]. However, the challenge for using SBS is that producing high-quality semantic segmentation is time consuming and that the best semantic networks do not process video frames in real time [17].

In this work, we propose a novel algorithm, called Real-Time Semantic Background Subtraction (RT-SBS) and illustrated in Figure 1, which is capable to use semantic information provided at a slower pace, and for some pixels. It reuses previous semantic information by integrating a change detection algorithm during the decision process. The latter checks if the last decision enforced by the semantic information is still up to date to be replicated. This allows our algorithm to keep performances close to the ones of SBS, while being real time. Furthermore, we introduce a semantic feedback to further improve the performance of SBS.

The paper is organized as follows. Section 2 presents SBS under a novel point of view which then allows to introduce RT-SBS. Then, in Section 3, we show state-of-the-art performance of our algorithm on the CDNet 2014 dataset and compare it to several other algorithms. Finally, we conclude the paper in Section 4.

Contributions. We summarize our contributions as follows.

- (i) We present a novel real-time background subtraction algorithm that leverages high-quality semantic information which can be provided at any pace and independently for each pixel.
- (ii) We show state-of-the-art performance on the CDNet 2014 dataset with our real-time algorithm.

2. REAL-TIME SEMANTIC BACKGROUND SUBTRACTION

Recently, Braham *et al.* have introduced the Semantic Background Subtraction (SBS) algorithm [14] that leverages semantic information to help addressing the challenges of background subtraction. One major constraint with SBS is that it requires reliable semantic information and that the best current networks are far from a real-time frame rate. Hence, SBS cannot be used in real-time constrained applications. In this work, we propose a novel background subtraction algorithm that extends SBS for real-time applications, regardless of the semantic network processing speed.

2.1. Description of semantic background subtraction

SBS combines the decision of two classifiers that operate at each pixel (x, y) and for each frame (indexed by t): (1) a background subtraction algorithm, which is a binary classifier between the background (BG) and the foreground (FG), whose output is denoted by $B_t(x, y) \in \{\text{BG}, \text{FG}\}$, and (2) a semantic three-class classifier, whose output is denoted by $S_t(x, y) \in \{\text{BG}, \text{FG}, "?\}$, where the third class, called the

<i>Decision table of SBS</i>			
Classifiers			Output
	$B_t(x, y)$	$S_t(x, y)$	$D_t(x, y)$
(L1)	BG	"?"	BG
(L2)	BG	BG	BG
(L3)	BG	FG	FG
(L4)	FG	"?"	FG
(L5)	FG	BG	BG
(L6)	FG	FG	FG

Table 1. Decision table for the output of SBS ($D_t(x, y)$) based on the output of two classifiers: a BGS algorithm ($B_t(x, y)$) and a semantic classifier ($S_t(x, y)$).

“don’t know” (“?”) class, corresponds to cases where the semantic classifier is not able to take a decision. This semantic classifier is built upon two signals that contribute to take a decision. The first one is the semantic probability that pixel (x, y) belongs to a set of objects most likely in motion. If this signal is lower than some threshold τ_{BG} , $S_t(x, y)$ is set to BG. The second signal is a pixelwise increment of semantic probability for a pixel (x, y) to belong to a moving object. When this signal is larger than another threshold τ_{FG} , $S_t(x, y)$ is set to FG. In all other cases, $S_t(x, y)$ is undetermined and is assigned a “don’t know” class, denoted “?” in the following.

Finally, the output of SBS, noted $D_t(x, y)$, is a combination of $B_t(x, y)$ and $S_t(x, y)$, as outlined in Table 1. This combination works as follows: when $S_t(x, y)$ is determined (either BG or FG), this class is chosen as the output of SBS regardless of the value of $B_t(x, y)$; when $S_t(x, y)$ is undetermined (which corresponds to “?” cases), the class of $B_t(x, y)$ is chosen as $D_t(x, y)$.

While SBS is effective to handle challenging BGS scenarios, it can only be real time if both classifiers are real time. As the decision of the semantic classifier supersedes that of $B_t(x, y)$ in two scenarios (see lines (L3) and (L5) in Table 1), it is essential to rely on a high-quality semantic segmentation, which is not achievable with faster semantic networks.

Another way to reduce the computation time of semantic information is to segment small portions of the image or to skip some frames. However, according to the original decision table of SBS, this would introduce more “don’t know” cases for the semantic classifier (equivalent to lines (L1) and (L4) of Table 1). Our algorithm aims at providing a decision different from the “don’t know” case when the semantic segmentation has not been calculated for pixel (x, y) at time t .

2.2. Change detection for replacing missing semantic information

We propose a novel algorithm that reuses previous decisions of the semantic classifier in the absence of semantic information. We choose to rely on previously available semantic information and we check whether or not this information is

Decision table of RT-SBS				
Classifiers				Output
	$B_t(x, y)$	$S_{t^*}(x, y)$	$C_t(x, y)$	$D_t(x, y)$
(L1)	BG	"?"	"✖"	BG
(L2)	BG	BG	"✖"	BG
(L3)	BG	FG	No Change	FG
(L4)	BG	FG	Change	BG
(L5)	FG	"?"	"✖"	FG
(L6)	FG	BG	No Change	BG
(L7)	FG	BG	Change	FG
(L8)	FG	FG	"✖"	FG

Table 2. Decision table of RT-SBS. Its output ($D_t(x, y)$) depends on three classifiers: a BGS algorithm ($B_t(x, y)$), information about the last time, $t^* \leq t$, the semantic classifier ($S_{t^*}(x, y)$) classified the pixel, and a change detection algorithm ($C_t(x, y)$). The “don’t care” values (“✖”) represent cases where $C_t(x, y)$ has not impact on $D_t(x, y)$, either because previous semantic information is undetermined or because $B_t(x, y)$ and $S_{t^*}(x, y)$ agree on the class.

still relevant. If the pixel has not changed too much, its predicted semantic class is still likely untouched, and therefore the previous decision of the semantic classifier is replicated.

Technically, we introduce a third classifier in the previous decision table. This classifier corresponds to a change detection algorithm whose task is to predict whether or not a pixel’s value has significantly changed between the current image, at time t , and the last time semantic information was available for that pixel, at time $t^* \leq t$. The new decision table is presented in Table 2 and works as follows: if the change detection algorithm, whose output is denoted by $C_t(x, y)$, predicts that the pixel has not changed, it means that the pixel still probably belongs to the same object and thus the previous semantic decision is repeated. Alternatively, when the change detection algorithm predicts that the pixel has changed too much, the previous semantic information cannot be trusted anymore, leaving it to the BGS algorithm to classify the pixel. The improvement of our algorithm compared to SBS originates from lines (L3) and (L6) of Table 2, as without the change detection classifier, the final decision would be taken by the BGS algorithm alone.

The only requirement for the choice of the change detection algorithm is that it has to be real time. In RT-SBS, we choose a simple yet effective algorithm that relies on the Manhattan distance between the current pixel’s color value and its previous color value when semantic information was last available, at time t^* . If this color distance is smaller (resp. larger) than some threshold, the change detection algorithm predicts that the pixel has not changed (resp. has changed). Let us note that we use two different thresholds depending on the output of the semantic classifier (τ_{BG}^* if $S_{t^*}(x, y) = \text{BG}$, or τ_{FG}^* if $S_{t^*}(x, y) = \text{FG}$) since the foreground objects and

the background change at different rates. In the case where semantic information is available, the change detection algorithm will obviously always predict that the pixel has not changed since $t^* = t$, and the decision table of RT-SBS (Table 2) degenerates into that of SBS (Table 1).

2.3. Introducing a semantic feedback

The last choice to make is the one of a real-time BGS algorithm. We choose to use ViBe [18] as it is the best real-time BGS algorithm according to [10]. This algorithm has the particularity of updating its background model in a conservative way, meaning that only pixels classified as background (or close to a background pixel) can be updated. Instead of keeping the output of ViBe for the update ($B_t(x, y)$), we replace it with the output of RT-SBS ($D_t(x, y)$) which is better. This introduces a feedback of the semantic information in the background model (via lines (L3) and (L6) of Table 2), which makes ViBe take better decisions at each new frame. It is interesting to note that RT-SBS could be used with any other BGS algorithm, just like SBS, and that the semantic feedback is an add-on in the case of ViBe.

3. EXPERIMENTAL RESULTS

For the evaluation, we choose the CDNet 2014 dataset [5]. It is composed of 53 video sequences mostly shot at 25 fps and comprising some challenging scenarios such as intermittent object motion, dynamic backgrounds and moving cameras. We use the overall F_1 score as suggested in the evaluation policy to compare RT-SBS with the other BGS algorithms. Semantic segmentation is computed as in [14] using the semantic segmentation network PSPNet [15]. We consider that semantic information is available for one in every X frame, which is denoted as $X:1$. As PSPNet runs at about 5 frames per second (fps) on the images of the dataset according to [14], only a maximum of 1 out of 5 frames can have access to the semantic information in a real-time setup, which is denoted as 5:1. In our experiments, the performance is computed for several semantic frame rates $X:1$. Real-time configurations correspond to $X \geq 5$.

The parameters of RT-SBS (τ_{BG} , τ_{FG} , τ_{BG}^* , τ_{FG}^*) are optimized through a Bayesian optimization process [19] on the entire dataset for each X with the overall F_1 score as optimization criterion. Let us note that the case $X = 1$ corresponds to SBS as semantic information is available for all frames. To show the importance of repeating the decision of the semantic classifier only when relevant, we compare our algorithm with two heuristics that can also extend SBS in the case of missing semantic information. The first heuristic never repeats the decision of the semantic classifier and the second heuristic always repeats the decision of the semantic classifier without checking if the pixel’s value has changed. Note that the former corresponds to RT-SBS with $\tau_{BG}^* < 0$

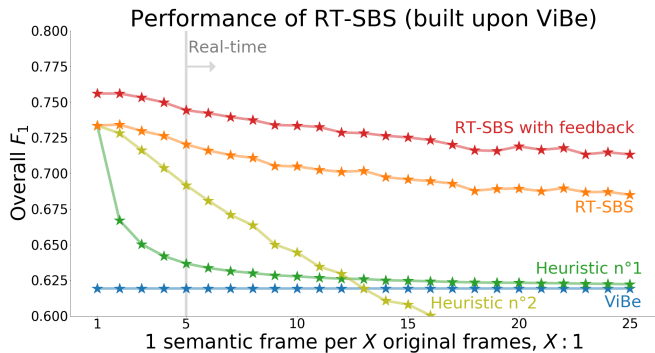


Fig. 2. Overall F_1 scores obtained with RT-SBS (built upon ViBe) as a function of the semantic frame rate $X:1$ of **RT-SBS with feedback**, **RT-SBS without feedback**, the **first heuristic**, the **second heuristic**, and the **original ViBe algorithm**.

and $\tau_{FG}^* < 0$ and the latter with τ_{BG}^* and τ_{FG}^* chosen larger than the upper bound of color distances.

Figure 2 reports the performances of RT-SBS built upon the ViBe BGS algorithm. We draw five important observations from this graph: (1) RT-SBS always improves the performance compared to ViBe, even when the semantic frame rate is low. (2) Its best real-time performance (at 5:1) is very close to the one of SBS (at 1:1, without feedback). (3) Both heuristics perform way worse than RT-SBS, indicating that the semantic information should only be repeated when relevant. This points out the importance of the selectivity process introduced by the change detection algorithm in RT-SBS. (4) Including a semantic feedback improves the performance for all semantic frame rates. This is because the internal model of the BGS algorithm is improved, and thus its decisions are better overall. Even at 5:1, our algorithm with feedback surpasses the performance of SBS. (5) The best real-time performance is achieved with the feedback at 5:1 with a F_1 score of 0.746. We compare this score with the top-5 state-of-the-art unsupervised background subtraction algorithms in Table 3. As can be seen, the performance of our algorithm are comparable with the state-of-the-art algorithms which are not real time. Furthermore, our algorithm performs better than all real-time BGS algorithms, making it the state of the art for real-time unsupervised algorithms.

Also, we performed a scene-specific Bayesian optimization [19] of the parameters; this leads to one set of parameters for each video. It corresponds to a more practical use of a BGS algorithm where its parameters are tuned for each application. With this particular optimization, we obtain an overall F_1 score of 0.828. This high score should not be compared with the others, but still shows the great potential of our algorithm in real-world applications. Finally, we display some results in Figure 3 showing our improvements qualitatively.

Unsupervised BGS algorithms	F_1	fps
SemanticBGS (SBS with IUTIS-5) [14]	0.789	≈ 7
IUTIS-5 [9]	0.772	≈ 10
IUTIS-3 [9]	0.755	≈ 10
WisenetMD [20]	0.754	≈ 12
WeSamBE [21]	0.745	≈ 2
PAWCS [22]	0.740	$\approx 1 - 2$
ViBe [7]	0.619	≈ 152
RT-SBS at $X : 5$	0.746	25
RT-SBS at $X : 10$	0.734	50
RT-SBS at $X : 5$ and scene-specific optimization	0.828	25

Table 3. Comparison of the performance and speed of RT-SBS (build upon ViBe + feedback) with the top-5 unsupervised BGS algorithms on the CDNet 2014 dataset and the previous best real-time one. Our algorithm improves on some state-of-the-art algorithms while being real time and surpasses all real-time BGS algorithms. The mean frame rates (fps) are taken from [23].

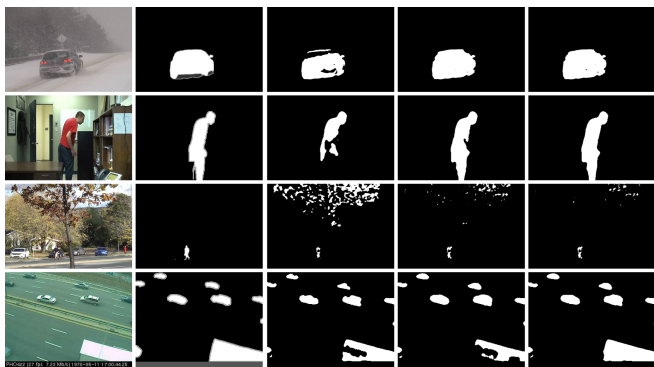


Fig. 3. Qualitative evaluation of RT-SBS. From left to right: the original color image, the ground truth, the background subtraction of ViBe, RT-SBS, and RT-SBS with a feedback.

4. CONCLUSION

We presented a novel background subtraction algorithm, called Real-Time Semantic Background Subtraction (RT-SBS), that extends the semantic background subtraction (SBS) algorithm for real-time applications. RT-SBS leverages high-quality semantic information which can be provided at any pace and independently for each pixel and checks its relevance through time using a change detection algorithm. We showed that our algorithm outperforms real-time background subtraction algorithms and competes with the non-real-time state-of-the-art ones. Python CPU and GPU codes of RT-SBS are available at <https://github.com/cioppaanthony/rt-sbs>.

Acknowledgment. A. Cioppa has a grant funded by the FRIA, Belgium. This work is part of a patent application (US 2019/0197696 A1).

5. REFERENCES

- [1] T. Bouwmans, “Traditional and recent approaches in background modeling for foreground detection: An overview,” *Computer Science Review*, vol. 11-12, pp. 31–66, May 2014.
- [2] C. Stauffer and E. Grimson, “Adaptive background mixture models for real-time tracking,” in *IEEE Int. Conf. Comput. Vision and Pattern Recogn. (CVPR)*, Fort Collins, Colorado, USA, June 1999, vol. 2, pp. 246–252.
- [3] A. Elgammal, D. Harwood, and L. Davis, “Non-parametric model for background subtraction,” in *Eur. Conf. Comput. Vision (ECCV)*, June 2000, vol. 1843 of *Lecture Notes Comp. Sci.*, pp. 751–767, Springer.
- [4] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequière, “A benchmark dataset for outdoor foreground/background extraction,” in *Asian Conf. Comput. Vision (ACCV)*, Nov. 2012, vol. 7728 of *Lecture Notes Comp. Sci.*, pp. 291–300, Springer.
- [5] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, “CDnet 2014: An expanded change detection benchmark dataset,” in *IEEE Int. Conf. Comput. Vision and Pattern Recogn. Workshops (CVPRW)*, Columbus, Ohio, USA, June 2014, pp. 393–400.
- [6] C. Cuevas, E. Yanez, and N. Garcia, “Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA,” *Comp. Vision and Image Understanding*, vol. 152, pp. 103–117, Nov. 2016.
- [7] O. Barnich and M. Van Droogenbroeck, “ViBe: A universal background subtraction algorithm for video sequences,” *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, June 2011.
- [8] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, “Universal background subtraction using word consensus models,” *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4768–4781, Oct. 2016.
- [9] S. Bianco, G. Ciocca, and R. Schettini, “Combination of video change detection algorithms by genetic programming,” *IEEE Trans. Evolutionary Computation*, vol. 21, no. 6, pp. 914–928, Dec. 2017.
- [10] S. Roy and A. Ghosh, “Real-time adaptive Histogram Min-Max Bucket (HMMB) model for background subtraction,” *IEEE Trans. Circuits and Syst. for Video Technol.*, vol. 28, no. 7, pp. 1513–1525, July 2018.
- [11] M. Braham and M. Van Droogenbroeck, “Deep background subtraction with scene-specific convolutional neural networks,” in *IEEE Int. Conf. Syst., Signals and Image Process. (IWSSIP)*, May 2016, pp. 1–4.
- [12] B. Garcia-Garcia, T. Bouwmans, and A. J. Rosales Silva, “Background subtraction in real applications: Challenges, current models and future directions,” *Comp. Sci. Review*, vol. 35, pp. 1–42, Feb. 2020.
- [13] W.-B. Zheng, K.-F. Wang, and F.-Y. Wang, “Background subtraction algorithm with Bayesian generative adversarial networks,” *Acta Automatica Sinica*, vol. 44, no. 5, pp. 878–890, May 2018.
- [14] M. Braham, S. Piérard, and M. Van Droogenbroeck, “Semantic background subtraction,” in *IEEE Int. Conf. Image Process. (ICIP)*, Beijing, China, Sept. 2017, pp. 4552–4556.
- [15] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *IEEE Int. Conf. Comput. Vision and Pattern Recogn. (CVPR)*, Honolulu, HI, USA, July 2017, pp. 6230–6239.
- [16] M. Tezcan, P. Ishwar, and J. Konrad, “BSUV-Net: A fully-convolutional neural network for background subtraction of unseen videos,” in *IEEE Winter Conf. Applicat. Comp. Vision (WACV)*, Snowmass village, Colorado, USA, Mar. 2020, pp. 2774–2783.
- [17] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: A survey,” *CoRR*, vol. abs/2001.05566v1, 2020.
- [18] O. Barnich and M. Van Droogenbroeck, “ViBe: a powerful random technique to estimate the background in video sequences,” in *Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, Apr. 2009, pp. 945–948.
- [19] “Bayesian optimization,” <https://github.com/fmfn/BayesianOptimization>, Last accessed: 2019-10-20.
- [20] S. h. Lee, G. c. Lee, J. Yoo, and S. Kwon, “WisenetMD: Motion detection using dynamic background region analysis,” *Symmetry*, vol. 11, no. 5, pp. 1–15, May 2019.
- [21] S. Jiang and X. Lu, “WeSamBE: A weight-sample-based method for background subtraction,” *IEEE Trans. Circuits and Syst. for Video Technol.*, vol. 28, no. 9, pp. 2105–2115, Sept. 2018.
- [22] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, “A self-adjusting approach to change detection based on background word consensus,” in *IEEE Winter Conf. Applicat. Comp. Vision (WACV)*, Waikoloa Beach, Hawaii, USA, Jan. 2015, pp. 990–997.
- [23] P.-M. Jodoin and J. Konrad, “Change detection website,” <http://changedetection.net/>.