Universal Background Subtraction Using Word Consensus Models

Pierre-Luc St-Charles, *Student Member, IEEE,* Guillaume-Alexandre Bilodeau, *Member, IEEE,* Robert Bergevin, *Member, IEEE*

***Abstract*—Background subtraction is often used as the first step in video analysis and smart surveillance applications. However, the issue of inconsistent performance across different scenarios due to a lack of flexibility remains a serious concern. To address this, we propose a novel non-parametric, pixel-level background modeling approach based on word dictionaries that draws from traditional codebooks and sample consensus approaches. In this new approach, the importance of each background sample (or word) is evaluated online based on their recurrence among all local observations. This helps build smaller pixel models that are better suited for long-term foreground detection. Combining these models with a frame-level dictionary and local feedback mechanisms leads us to our proposed background subtraction method, coined "PAWCS". Experiments on the 2012 and 2014 versions of the ChangeDetection.net dataset show that PAWCS outperforms 26 previously tested and published methods in terms of overall F-Measure as well as in most categories taken individually. Our results can be reproduced with a C++ implementation available online.**

***Index Terms*—video segmentation, word consensus, change detection, background subtraction, video signal processing.**

I. Introduction

The segmentation of foreground and background regions in video sequences based on change detection is a fundamental, yet challenging early vision task. Often simply called background subtraction, it has been well studied over the years. It generally serves as a low cost, high accuracy alternative to unconstrained binary segmentation based on spatiotemporal feature clustering. Background subtraction is typically based on a single hypothesis: all images of a sequence share a common "background", from which discrepancies are to be considered of interest (or "foreground"). Thus, it is especially useful in applications with static cameras, or when registration between images is possible, as background modeling and foreground classification can be solved solely at the pixel level. Furthermore, this type of segmentation requires no prior knowledge of the foreground, making it ideal for online surveillance applications (or more generally, in intelligent environments).

The main challenges of background subtraction lie in adaptive background modeling and in the definition of "relevant change", i.e. deciding how discrepancies between observations and model predictions should be classified. In nearly all applications, the background cannot be considered timeless as it may present noisy or dynamic elements (e.g. rippling water, swaying trees), and its content may change over the sequence (e.g. cars entering and leaving a parking lot). Also, while easily detectable changes caused by illumination variations may not be relevant to most applications, subtle changes caused by "camouflaged" foreground objects (i.e. similar to the background) have to be detected correctly. Classic and modern background subtraction challenges have been highlighted in [1]–[4].

Research has previously focused on improving modeling and classification for selected challenges individually, but very little work has been addressing them holistically. Therefore, most background subtraction methods require significant application-specific tuning to achieve good segmentation results in complex scenarios. Few of them actually perform well across many common use cases without supervision or preprocessing. A "universal" background subtraction solution has to:

1) learn the proper balance between sensitivity and precision based on past observations and segmentation coherence to make good unsupervised decisions;
2) ignore irrelevant changes in the observed scene which concur with previously recognized patterns; and
3) determine how and when foreground objects are absorbed in the background model, and avoid model corruption when the background is altered.

Achieving these objectives is complicated by the nature of most background subtraction approaches that, by design, operate online at the pixel level for better efficiency. As such, they cannot easily analyze large-scale change patterns, and must rely on complex regularization schemes (e.g. frame-wide energy minimization with higher order potentials) to produce good results.

What we propose in this paper is a background subtraction method that can be applied to a large variety of scenarios without manual parameter readjustment, coined PAWCS (Pixel-based Adaptive Word Consensus Segmenter). More specifi-

Manuscript received XXXX XX, 2016; accepted XXXX XX, 2016. Date of publication xxxxxxxx xx 2016; date of current version xxxxxxxxx xx 2016. This work was supported in part by NSERC, FRQ-NT team grant No. 2014-PR-172083, and by REPARTI (Regroupement pour l'étude des environnements partagés intelligents répartis) FRQ-NT strategic cluster. The associate editor coordinating the review of this manuscript and approving it for publication was xxxxxxxxxxxx.

P.-L. St-Charles and G.-A. Bilodeau are with the Laboratoire d'Interprétation et de Traitement d'Images et Vidéo (LITIV), Polytechnique Montréal, Montréal, QC, Canada. (e-mail: pierre-luc.st-charles@polymtl.ca; gabilodeau@polymtl.ca).

R. Bergevin is with the Laboratoire de Vision et Systèmes Numériques (LVSN), Université Laval, Québec City, QC, Canada. (e-mail: robert.bergevin@gel.ulaval.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier XX.XXXX/XXX.XXXX.XXXXXXX

cally, we first introduce a new persistence-based word dictionary scheme for instance-based background modeling that simultaneously addresses short-term and long-term adaptation challenges at the pixel and frame level. Unlike traditional codebook or sample consensus approaches, this novel non-parametric modeling strategy allows for the online principled learning of static and dynamic background regions at a low memory cost. This is because it dynamically maintains the minimal number of background samples (or words) required for proper segmentation. Persistence estimation is used to gauge the importance and reliability of each background word over time based on local match counts. Persistence values then influence the rate at which each word is updated and used for classifications. The long-term retention of good words despite the presence of static foreground and the rapid suppression of irrelevant words (e.g. captured while bootstrapping) are thus assured by design. In other words, PAWCS requires no explicit training to populate its background models, and keeps them up-to-date while processing new frames.

Our novel word models adopt the primary update and maintenance principles of stochastic sample consensus models, meaning that background words:1) do not need to be unique inside dictionaries; 2) can be built upon local image descriptors; and 3) are randomly updated in an online fashion. This strategy diverges from the traditional codebook maintenance strategy (i.e. [5]) as it allows words to overlap in feature space. This also means words can be shared between pixel-level dictionaries and can contribute to a frame-wide dictionary without having to solve costly unicity conflicts. As discussed in [6]–[8], spreading information between neighboring pixel models drastically improves segmentation coherence since it acts as a regularization step; the same is true for our method. The use of a frame-wide ("global") dictionary as a complement to pixel-level ("local") dictionaries further improves spatial coherence, and allows the capture of large-scale background change patterns.

Our proposed method also automatically adjusts its primary parameters by incorporating closed-loop controllers into each pixel-level model. That way, each background region can exhibit its own modeling and classification behavior, which can also evolve over the analyzed sequences. Parameter adjustments are regulated by monitoring:1) segmentation noise prior to regularization; 2) similarity between background models and observations; 3) region instability based on recurring label changes; and 4) the propagation of illumination updates in local neighborhoods. These four aspects are used to guide and/or trigger various feedback mechanisms that affect model update rates, matching and classification thresholds. Unlike previous methods that could also dynamically adapt to the scene (e.g. [9], [10]), our strategy is not hindered by foreground, and it does not rely on a sliding window analysis of inputs/outputs. Instead, it relies on a causal infinite impulse response filter and simple heuristics to quickly and efficiently react to short-term and intermittent disturbances at the pixel level.

We evaluate PAWCS using the 2012 and 2014 versions of the ChangeDetection.net (CDnet) benchmark [1], [11], [12] and compare our results with those of 21 methods listed on its online platform as well as the self-reported results of 6 recently published methods (27 in total). Our new approach outperforms most of them in overall performance, and in most categories taken individually. To make its usage and future comparisons outside this benchmark easier, we offer the full C++ implementation of PAWCS along with its entire testing framework online[1].

Note that our method was previously introduced in [13]; here, we offer an extended description of our approach, discuss new experiments on the 2014 CDnet dataset, and compare our results with the new state-of-the-art.

II. Related Work

Many background modeling paradigms have been introduced over the years: the earliest and most popular is pixel-level modeling, as it allows simple, scalable, high-speed implementations. This is due to the fact that this modeling strategy relies on low-level features (e.g. color intensities, gradients) to track background representations, as opposed to region-level or object-level information. Two of the three best methods listed in [11], namely PBAS [10] and ViBe+ [14], follow this paradigm as they are both based on stochastic intensity sampling [6]. Parametric approaches based on Gaussian Mixture Models (GMM) [15] or non-parametric ones based on Kernel Density Estimation (KDE) [16] as well as their derivatives [9], [17] can all be considered part of this family. With a goal similar to [9], Haines and Xiang [18] recently proposed a Dirichlet process Gaussian mixture modeling approach that automatically determines its optimal distribution parameters and component count in a data-driven fashion. Their "confidence capping" strategy allows background representations to be captured and forgotten in a principled manner, much like the "maximum negative run-length" evaluation strategy proposed by Kim et al. [19]. In this latter work, codewords are introduced, which are created by clustering reoccurring background representations during a training phase. These codewords are then amassed into codebooks at the pixel-level to create independent, low-level background models. Compared to classic non-parametric models, codebooks can accurately describe multimodal background regions and avoid corruption due to static foreground while having a small memory footprint. This modeling approach was first improved in [5] to allow the online adaptation of codebooks, then in [20] by extending codewords to the spatiotemporal domain, and more recently in [21] by using duration dependent hidden Markov models to identify and capture periodic background change patterns. Our proposed word-based modeling approach detailed in the following section differs from traditional codebook approaches in that it does not cluster background representations into unique codewords during a training phase. Furthermore, it allows words to be sorted within background models and replaced online based on their persistence in the observed data.

Pixel-based methods are not restricted to use intensity values for background modeling: they can also capture texture information using local descriptors. Doing so helps produce richer, "spatially aware" background samples that are critical

[1] https://github.com/plstcharles/litiv

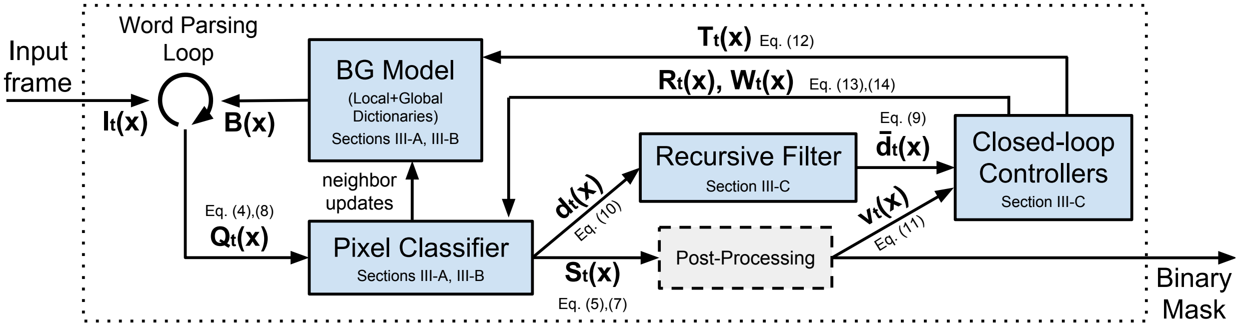


Fig. 1. Block diagram of the Pixel-based Adaptive Word Consensus Segmenter. Each block represent a major component detailed in Section III.

for the detection of camouflaged foreground objects. [22]–[26] all use Local Binary Patterns (LBP) or other similar binary features for this purpose and achieve good tolerance to illumination variations. The works of [2], [27] have studied the proper selection and usage of low-level features in pixel-level background modeling.

Frame-level background modeling via Principal Component Analysis (PCA) and low-rank/sparse decomposition approaches is a popular alternative to pixel-level modeling [28]–[31]. These approaches are however not ideal for surveillance applications as most rely on batch or offline processing or suffer from scaling problems. The authors of [32] address scaling problems by reformulating principal component analysis for 2D images, and their method achieves much lower memory consumption and computational cost than traditional methods. Some online approaches have also been proposed recently [33]–[35], but they are still very computationally expensive.

An early finding in the field is that background modeling should not be limited in scope to frame-wide or pixel-independent processes [36]. Doing so would restrict their perception of change to a single spatial scale and therefore make it harder to address common background maintenance challenges. Multi-scale and "hybrid" methods such as those of [24], [36]–[38] have emerged in light of this to improve segmentation coherence at different spatial scales. Coarse-to-fine and block-based strategies can also be adopted to solve this problem. For example, Staglianò et al. [39] recently proposed a block-based modeling approach based on sparse coding that relies on patch dictionaries learned online to improve spatial coherence. Furthermore, features extracted from High-Efficiency Video Coding (HEVC) macroblocks have been used in [40] for low-cost block-based modeling with good spatiotemporal coherence. Alternative solutions that rely on self-organizing maps [41]–[43], neural networks [44], [45], or probabilistic frameworks such as Markov Random Fields [20], [46], [47] to expose relationships between independent background models have also been studied.

A problem common to most classic methods is that they lack flexibility: even though they can provide good results on individual sequences when tuned properly, few of them can actually perform equally well across large datasets when used "out-of-the-box". Modern methods sometimes propose ways to dynamically control either model complexity [9], [18], adaptation rates [48], [49] or classification thresholds [14], [50] online. These however often rely on complex object-level or frame-level analyses and react slowly to intermittent changes. In the case of PBAS [10], which simultaneously controls model adaptation rates and classification thresholds via pixel-level feedback loops, delayed and fluctuating sensitivity variations can still cause segmentation problems. Our previous work [8] addressed this by proposing fast-response feedback loops based on a dual measurement approach, but did not offer a way to simultaneously control model complexity. The recent solution of [51] also addresses the flexibility problem of segmentation methods by combining them using a genetic programming approach.

The interested reader is referred to recent surveys [1]–[3], [52], [53] for details on the background subtraction field.

III. Methodology

As shown in Figure 1, our proposed method can be split into five components:1) a full "background model", actually composed of multiple pixel-level (local) models and a frame-level (global) model; 2) a classifier, which produces "raw" segmentation decisions for each pixel based on outlier detection; 3) a post-processing (or regularization) step that relies on basic morphological operations to measure and eliminate segmentation noise; 4) a recursive measurement filter used to simulate the response of a sliding window over model-observation similarity indicators; and 5) a closed-loop controller block, responsible for adjusting the internal parameters of other components based on their state and output. The pixel- and frame-level models along with our classification strategy are presented in Sections III-A and III-B. The recursive measurement filter as well as feedback mechanisms and controllers are discussed in Section III-C. Finally, three heuristics adopted to further improve our method's adaptability are presented in Section III-D.

A. *Word consensus for pixel-level modeling*

Our novel non-parametric modeling approach is essentially a hybrid between codebook [5] and sample consensus [7], [54] strategies. "Word consensus" inherits the main advantages of these modeling strategies while avoiding their pitfalls, namely costly updates and high memory requirements. As in other pixel-level, non-parametric modeling approaches, the idea behind word consensus is to simultaneously build independent background models by gathering data samples from local observations. Then, new observations can be classified based on their model overlap. In the following paragraphs, we present an overview of our proposed word-based modeling approach and define some basic terms, and then discuss feature matching, classification and update mechanisms.

Overview and definitions. In contrast to the terminology of [5], we consider model samples as "words" instead of "codewords" since they are not obtained via clustering, and are thus not necessarily unique. Furthermore, our word-based models are termed "dictionaries" instead of "codebooks", as the words they contain are sorted and systematically parsed for matches during classification. We define the "local" dictionary of a given pixel x as

$$B_l(x) = \{\omega_1, \omega_2, \dots, \omega_N\} \tag{1}$$

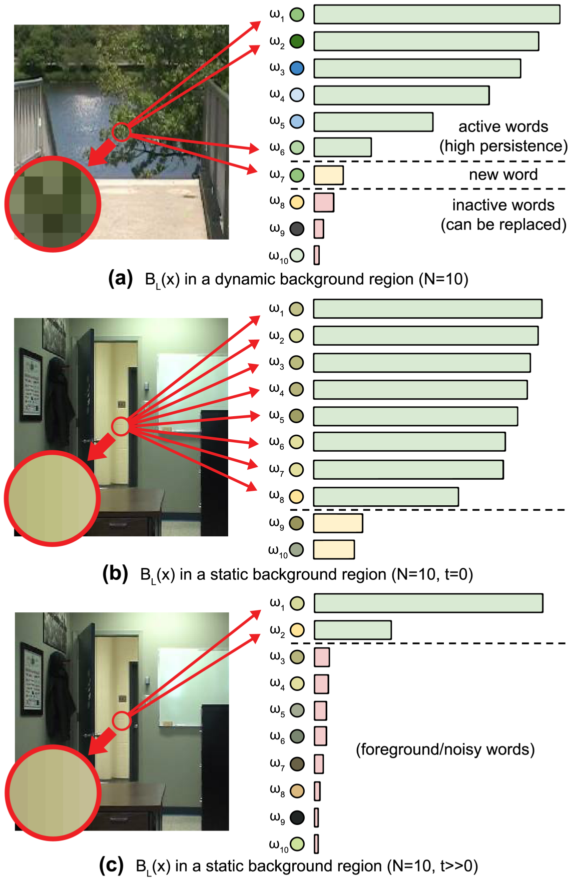
where ω_n are background words, and N is the number of words in the dictionary. Each word essentially consists of a background representation (characterized using local image descriptors and/or low-level features) and a transient persistence value. This value is estimated online based on the recurrence (i.e. match count) of the word among recent observations at x. In our modeling approach, persistence defines the basic criteria of word replacement and update policies: the more "persistent" a word is, the less likely it is to be forgotten or replaced by new observations. In comparison to modeling approaches with planned obsolescence policies, our approach is especially useful when segmenting intermittently moving objects.

We evaluate the persistence of a word ω at time t using

$$q_t(\omega) = \frac{n_\omega}{(t^l_\omega - t^{\mathrm{f}}_\omega) + 2 \cdot (t - t^l_\omega) + t_0}, \tag{2}$$

where n_ω is ω's total occurrence count, t^{f}_ω and t^l_ω are, respectively, the time at which it was first and last observed, and t_0 is a fixed offset value. The principle behind this equation unique to our modeling approach is comparable to the maximum negative run-length measure of [5] and the confidence capping of distribution components in [18]: it promotes the retention of recurring background words and helps forget those that have not been observed recently. The first denominator term, $(t^l_\omega - t^{\mathrm{f}}_\omega)$, measures the lifespan of ω and is used to scale the persistence of words to the $[0, 1]$ range. It also reflects the "persistence inertia" of ω, i.e. how well it resists persistence value fluctuations caused by short time spans without occurrences. Besides, note that since the second term of the denominator, $(t - t^l_\omega)$, is multiplied by two, the penalty incurred by a distant last occurrence is essentially doubled. This means that while recurrence is important, good words that suddenly disappear from observations can still be eliminated quickly despite very long lifespans. The time offset t_0 is only used here to prevent new words from having important persistence values.

Given (2), a mandatory training phase to learn and filter words in local dictionaries is not required since the importance of all words can be continuously estimated over the analyzed sequence. Thus, the segmentation process can be fully initialized by sampling local pixel neighborhoods from a single video frame that may contain foreground. Dictionaries will then naturally stabilize over time to contain only recurring background words. This approach however requires proper feature matching and update mechanisms, which will be discussed next. For now, note that our local dictionaries adapt to permanent background changes by learning a new word when local observations cannot be matched to any existing background word. We illustrated the content of a local dictionary for a multimodal background region in Figure 2a.



(a) B_L(x) in a dynamic background region (N=10)

(b) B_L(x) in a static background region (N=10, t=0)

(c) B_L(x) in a static background region (N=10, t>>0)

Fig. 2. Illustrations of possible local dictionary content for dynamic and static background regions. The bars next to each word represents their relative importance in the dictionary based on persistence. In (a), different words are kept active simultaneously while new words are inserted. In (b), the dictionary is shown right after model initialization, and many overlapping words are present due to local neighborhood sampling. In (c), the same dictionary as (b) is presented but at a later time in the sequence, showing a reduced number of active words.

To avoid the infinite expansion of local dictionaries, we cap the number of active words they can contain (N). To keep word counts low, we can ignore or remove words that have negligible persistence values. If a dictionary is full, words with the lowest persistence values are eliminated and replaced by new ones. To make these evaluations and replacements more efficient, we systematically reorder the content of local dictionaries during the matching process using a bubble sort algorithm. That way, more persistent words are checked for matches first, and they are less likely to be replaced by new ones.

Feature selection and matching. In PAWCS, we characterize pixel neighborhoods (for both background words and local observations) using RGB intensities and Local Binary Similarity Pattern (LBSP) descriptors [55]. Local descriptors can be used like any other low-level feature in our modeling approach since, by design, we avoid merging or clustering data samples. The goal of using LBSP here is to improve robustness to illumination variations and enhance segmentation coherence (or smoothness) through local texture description and matching. As shown in [23], this color/LBSP description approach also boosts change detection sensitivity (which is beneficial), but induces additional false positives in regions with dynamic textures. We discuss how we solve this problem via feedback mechanisms below.

First, to determine if a word ω from a local dictionary $B_l(x)$ matches the observation of x at time t (noted $I_t(x)$), we evaluate their color resemblance (via ℓ_1 distance and color distortion [19]) and LBSP intersection (via Hamming distance). If all three distances fall below given change detection thresholds, then ω is considered a match for $I_t(x)$. To shorten the equations presented in the following sections, we simply write $||I_t(x) - \omega|| < R_t(x)$ to refer to this matching step. We define $R_t(x)$ as a distance threshold for x at time t, with $R_t(x) \geq 1$. In practice, color and LBSP distance thresholding is done in parallel, and their threshold values are respectively calculated from $R_t(x)$ using

$$R_{c,t}(x) = R_c \cdot \sqrt{R_t(x)} \tag{3a}$$

and

$$R_{d,t}(x) = R_d + 2^{R_t(x)} \ , \tag{3b}$$

where R_c and R_d are fixed baseline values. Unlike color thresholds, LBSP descriptor thresholds rely on an exponential relation which is better suited to their nature: small $R_{d,t}(x)$ values lead to discriminative texture matching, and larger $R_{d,t}(x)$ values are used for approximative gradient matching. As we will discuss in Section III-C, $R_t(x)$ is automatically increased in regions exhibiting dynamic textures. A large enough $R_t(x)$ value can thus exclude LBSP descriptors from the matching process by inducing a $R_{d,t}(x)$ value larger than the maximum LBSP Hamming Distance. This leads to a desirable reduction of change detection sensitivity, and ultimately to the elimination of false positives in dynamic texture regions.

Pixel classification. The nature of binary segmentation methods based on change detection implies that all pixels in a video frame are considered foreground unless their description matches with the background model. As explained earlier, we do not impose a unicity constraint on words in local dictionaries. This means that finding a single match in the background model for an observation is not enough to directly proclaim a pixel as background. Our classification approach instead relies on the idea of pixel labeling via consensus, i.e. we consider all words from a local dictionary that overlap (or match) the observation of a pixel in feature space to determine its segmentation label. In classic sample consensus methods, the number of matched background samples is the sole determinant of the classifier: if at least a given number of samples are matched, the pixel is labeled as background. This can be interpreted as one-class classification for outlier detection. In our case, rather than using a fixed match count as the background classification threshold, we calculate the persistence sum of all matched words, and compare this value with a second dynamic threshold (noted $W_t(x)$). More specifically, for a pixel x, we compute the local dictionary persistence sum as

$$Q_{l,t}(x) = \sum_{\omega \, \in \, B_l(x)} q_t(\omega) \ \text{ s.t. } \ ||I_t(x) - \omega|| < R_t(x), \tag{4}$$

and then classify x as foreground (1) or background (0) using

$$S_t(x) = \begin{cases} 1 & \text{if } Q_{l,t}(x) < W_t(x) \\ 0 & \text{otherwise} \end{cases} \ , \tag{5}$$

where S_t is the output (raw) segmentation map. Again, we discuss how $W_t(x)$ is automatically computed in Section III-C.

Since we do not rely on foreground modeling, (4) and (5) encapsulate a one-class outlier detector. Here, $R_t(x)$ dynamically controls the reachability distance in feature space, which is similar to having an adaptive k in a k-Nearest Neighbor classifier. On the other hand, $W_t(x)$ dictates the minimum cumulative instance weight (or persistence sum) required to consider x as an inlier. Note that the persistence sum $Q_{l,t}(x)$ can be seen as a general indicator of how well $I_t(x)$ is represented by the content of $B_l(x)$. We will reuse this representativeness estimation in Section III-C to evaluate model-observation similarity.

Remember that words are sorted in local dictionaries based on their persistence values; this means the summation in (4) can be implemented so that it stops once the accumulated persistence is greater than $W_t(x)$. The primary ensuing advantage is faster processing, as dictionaries do not need to be fully parsed for matches. This termination criterion however also has a beneficial effect on the way word persistence values fluctuate over time, which we discuss next.

Word and dictionary updates. As stated before, the persistence value of a word can be evaluated online using (2) based on its occurrence count and the time of its first/last observations. In practice, these persistence parameters are only updated when a match is found in (4). This means that if the summation stops before reaching a potential matching word ω because similar alternatives exist with higher persistence values, then ω's persistence will slowly decay, and it will slide down in the dictionary's word ranking. As illustrated in Figure 2b and 2c, this is actually an important update mechanism that gradually eliminates background words which overlap in feature space. This mechanism is thus responsible for controlling the number of active words (i.e. words with a significant persistence value) in each local dictionary based on the complexity of its associated background region. Below, we discuss two other update mechanisms needed to make our proposed modeling approach universal.

The modeling approach described so far cannot cope with gradual background changes (e.g. the illumination variation caused by a growing cloud cover) since the background descriptions kept by words inside local dictionaries are not updated. To improve robustness in such cases, we randomly replace the color component of matched background words with observed values. This is only done when local texture variation (expressed by the distance between matched and observed LBSP descriptors) and color distortion (as presented in [19]) are negligible, and with probability $\rho = 1/T_t(x)$. Here, $T_t(x) \geq 1$ is a dynamic update rate further discussed in Section III-C. This update mechanism is meant to only allow a small proportion of background words to be modified in response to gradual illumination changes. Therefore, words left untouched will continue modeling the previous background state when the change is only temporary. This strategy implicitly prevents incorrigible model drift and the saturation of local dictionaries by words whose descriptions differ in brightness, but not in local texture. The direct replacement of a word's color component by an observed value also helps diversify dictionary content, and it is less costly than merging them.

In order to improve the consistency between neighboring local dictionaries, our method shares an important trait with recent sample-consensus methods (e.g. [6], [8], [10], [23]): pixel-level background information diffusion. The diffusion process increases robustness to infrequent periodic change by sharing information between afflicted regions, and helps erode tenacious false positive segmentation blobs caused by "ghosts" in the background model (see [4] for an exact definition). In its original form (introduced in [6]), information diffusion grants pixel models adjacent to x a chance to have one of their samples randomly replaced by the description of $I_t(x)$, but only if x is classified as background. To achieve similar results in our novel modeling approach, when (5) returns $S_t(x) = 0$, we randomly select a neighbor of x and update the persistence of each word of its local dictionary that matches with $I_t(x)$. Again, the probability of doing this is $\rho = 1/T_t(x)$. Note that a beneficial effect of these updates is a persistence boost for words in static background regions, which helps detect camouflaged and immobile foreground objects. In contrast to the spatial context improvement proposed in [20], our strategy allows pixel models to resist disturbances caused by small background displacements and vibrating cameras because spatial information is shared proactively.

The three proposed update mechanisms allow our pixel models to behave like codebooks in static background regions. Once stable, each local dictionary requires only one or two highly persistent words (which are kept up to date during illumination changes) to provide good segmentation results. In dynamic background regions, our pixel models are akin to sample consensus models, where 20 to 30 different words can be active at once, all with low cohesion and persistence values. Our pixel-level modeling approach alone is however unable to recognize patterns that are too large or outside the influence of local information diffusion. For this, we introduce a frame-level modeling approach, described next.

B. Word consensus for frame-level modeling

Using a frame-wide or "global" dictionary (noted B_g) along with local dictionaries for classifications fulfills the requirement set by the final Wallflower principle [36], i.e. background models should take into account changes at differing spatial scales. Capturing large scale background patterns is crucial when the observed scene exhibits intermittent change over large areas, or when background elements are removed from it. Besides, our pixel description approach produces highly specific words that, if tied to intermittent background patterns, may be discarded before being observed enough times to build a strong persistence value. Our proposed global dictionary model acts as a long term memory that tracks the use of common words and spreads their influence beyond the reach of the pixel-level diffusion update mechanism. Its only purpose is to provide a safeguard against false positive detections by relabeling x (i.e. overruling $S_t(x) = 1$) when a "global" word with strong persistence is matched to its pixel-level observation.

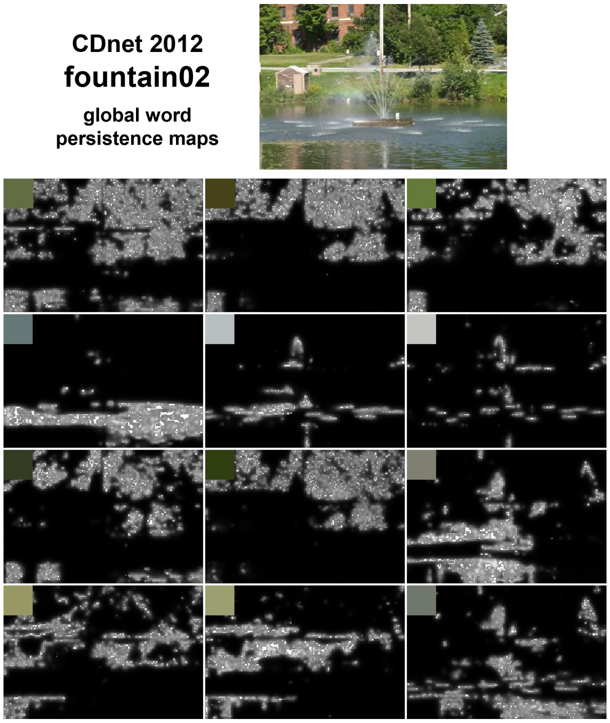


Fig. 3. Snapshots of the actual global word persistence maps used in the CDnet2012 fountain02 sequence at frame 500. The top-left corners show the pixel color description of each word; texture is omitted for illustration purposes. Brighter spots indicate where matches occurred and the map was updated; those points are the "seeds" from which the persistence is diffused. In total, 24 global words were active (the 12 with highest total persistence are shown here), covering over 95% of the image space with non-zero persistence values.

The persistence definition of (2) becomes problematic when considering background words in a global setting, as it implies no ownership or localization concept. Moreover, the state of a dictionary shared among all pixel models would depend on the ordering of local updates. Therefore, we define the persistence of global words by using 2D maps that "localize" the importance of these words in image space. These persistence maps can be seen as heat maps over the observed frames; some examples are shown in Figure 3. For each pixel x and each global word ω, a map value (noted $\Phi_\omega(x)$) is directly linked to a single pixel model $B_l(x)$. This means maps can be fully updated asynchronously. Furthermore, this one-for-one mapping in image space also means that frame-wide operations (blurring, saturation, decimation) on global word persistence maps are well defined and easy to implement.

Persistence map contents are updated in two ways. First, each time a pixel x is classified as background, it may be randomly elected to parse B_g for a matching word ω and perform a localized update of its persistence map. We define this update as

$$\Phi_\omega(x) = \min\big(\Phi_\omega(x) + Q_{l,t}(x), Q_{l,t}(x)\big). \tag{6}$$

This essentially creates a seed point in Φ_ω used to propagate persistence values to neighboring pixels. Second, the propagation itself is achieved by periodically blurring the maps of all global words using a normalized box filter. To avoid obtaining homogeneous results over time, $\Phi_\omega(x)$ values are also decimated while filtering. These two steps create another word diffusion mechanism that has a much larger reach than the one presented in Section III-A. Persistence map decimation also allows the global dictionary to "forget" words over time.

Finally, the foreground classification of a pixel x can be overruled if a matching global word (with a strong localized persistence) is found. More specifically, we replace (5) by

$$S_t(x) = \begin{cases} 1 & \text{if } Q_{l,t}(x) + Q_{g,t}(x) < W_t(x) \\ 0 & \text{otherwise} \end{cases}, \tag{7}$$

where

$$Q_{g,t}(x) = \begin{cases} \Phi_\omega(x) & \text{if } \exists\, \omega \in B_g \text{ s.t. } ||I_t(x) - \omega|| < R_t(x) \\ 0 & \text{otherwise (i.e. no match found)} \end{cases} \tag{8}$$

Since the LBSP descriptors we use are highly specific and hard to match across regions, we simplify the global word matching step in (8): instead of using a Hamming distance, we compare the Hamming weights of the descriptors. This is equivalent to an approximate gradient magnitude comparison. While gradient orientation information is lost, this approach preserves texture strength information, which is enough for discriminative change detection.

New words can also be inserted in global dictionaries: pixel models can be randomly selected to copy one of their highly-persistent local background words to B_g and initialize its persistence map, but only if it is missing from B_g. Again, the maximum number of active words in B_g is capped (N), and words with the lowest overall persistence are replaced first once full. The initialization is done by simply querying random pixel models for words and filling persistence maps until the global dictionary is full. Note that since (8) and the update mechanisms only target one word at a time (i.e. the first found match), and since global word descriptions do not change throughout their lifespan, feature space overlap between global words is highly unlikely. This means global dictionaries typically contain more unique words than local dictionaries.

C. Pixel-level feedback

Recall that the equations presented in the previous sections primarily relied on three types of pixel-independent parameters. Namely, the feature-space distance threshold used for matching in (4) and (8), $R_t(x)$; the persistence threshold used for classification in (5) and (7), $W_t(x)$; and the rate used to determine the probability of triggering all update mechanisms, $T_t(x)$. While fixed values could be used frame-wide, they would need to be determined empirically from the analyzed videos, and doing so would prevent pixel models from handling multiple segmentation challenges at once. Therefore, we dynamically adjust these internal parameters using pixel level closed-loop controllers. Subsequently, we use simple heuristics to increase the chance of triggering further model updates and parameter adjustments where needed. As stated in Section I, four aspects of our method are monitored and used to control these feedback mechanisms: two of them are related to the state of the background model (model-observation similarity and illumination update propagation) and two to the output (segmentation noise and instability). These are discussed below.

Evaluating the similarity between local dictionary words and local observations captured from input frames is the first step in determining if the background is adequately modeled. We do so for each pixel x by recursively filtering the estimations of minimal matching distances between local observations and words along with the "representativeness gap" of missed matches (defined below). This provides a temporally smooth measurement of model-observation similarity, noted $\bar{d}_t(x)$, which can be used in closed-loop controllers. The proposed recurrence relation is defined as

$$\bar{d}_t(x) = (1-\alpha)\cdot \bar{d}_{t-1}(x) + \alpha \cdot d_t(x) \tag{9}$$

with

$$d_t(x) = \max\left(\min_{\omega\in B_l(x)} ||I_t(x) - \omega||\,,\; \frac{W_t(x) - Q_{l,t}(x)}{W_t(x)}\right), \tag{10}$$

where $\alpha \in [0,1]$ is a fixed coefficient, and $\frac{W_t(x)-Q_{l,t}(x)}{W_t(x)}$ is the "representativeness gap". As stated earlier, this is because $Q_{l,t}(x)$ can be interpreted as an indicator of model representativeness.

In practice, since multiple distances are evaluated in $||I_t(x) - \omega||$, we find the minimum distance for each feature (color, LBSP) in $B_l(x)$ independently, normalize them to the $[0,1]$ range, and then only use the maximum one in (10). The "representativeness gap" is used to counterbalance the similarity score for models that have good matches but low persistence sums. In the end, $d_t(x)$ and $\bar{d}_t(x)$ can only take values in the $[0,1]$ range. The nature of the two terms compared in (10) makes "ideal" background modeling (reflected by $d(x)\approx 0$) very hard to achieve in practice. This forces continuous feedback that helps diversify pixel models. The recurrence relation of (9) essentially models the infinite impulse response of an exponentially weighted sliding window filter, but at a very low computational cost. This approach allows fast feedback responses to intermittent and irregular changes.

Since $\bar{d}_t(x)$ is updated every frame, foreground objects that pass or stay immobile over x will inevitably increase its value over time. Therefore, this first indicator cannot be used by itself to control parameter adjustments as it does not always truly reflect the similarity between background models and observations. To address this, we rely on a second pixel-level indicator, noted $v_t(x)$, that monitors noise in the segmentation results. The assumption behind $v_t(x)$ is that inadequately modeled regions emit more noise (i.e. alternating segmentation labels) than other regions, which are instead constantly labeled as foreground or background. We define $v_t(x)$ as a segmentation noise accumulator, and update it using

$$v_t(x) = \begin{cases} v_{t-1}(x) + 1 & \text{if } \big(S_t(x) \oplus S_{t-1}(x)\big) = 1 \\ v_{t-1}(x) - 0.1 & \text{otherwise} \end{cases} \tag{11}$$

where $\oplus$ is the XOR operator, and $v_t(x)$ is prevented from taking negative values. This definition essentially means that only static regions will exhibit low $v_t(x)$ values.

Using $\bar{d}_t(x)$ and $v_t(x)$, we can now describe our pixel-level controllers. First, we define the adjustment mechanism for local update rates as

$$T_t(x) = \begin{cases} T_{t-1}(x) + \dfrac{\lambda_T}{v_t(x) \cdot \bar{d}_t(x)} & \text{if } S_t(x) = 1 \\ T_{t-1}(x) - \dfrac{\lambda_T \cdot v_t(x)}{2 \cdot \bar{d}_t(x)} & \text{if } S_t(x) = 0 \end{cases} \tag{12}$$

where $\lambda_T \in [0, 1]$ is a fixed scaling factor, and $T_t(x)$ is bound to the $[1, 256]$ interval. Like all sample-consensus methods, we use an inversely proportional relation to calculate the probability ρ of triggering an update mechanism from $T_t(x)$. This means that high $T(x)$ values lead to fewer updates, and that when foreground is detected in static regions with low segmentation noise, model updates will almost immediately stop. In other words, $T(x)$ will max out quickly due to $v_t(x) \approx \bar{d}_t(x) \approx 0$. However, dynamic and noisy background regions will keep allowing model updates for much longer, as in those cases, $v_t(x) \gg 0$ and $\bar{d}_t(x) \approx 1$, which results in smoother variations.

Having models that update frequently is usually not enough to eliminate all false foreground classifications caused by strong dynamic background change. Locally adjusting feature matching thresholds is often the fallback solution to avoid having to build and maintain very large background models. As stated in Section III-A, we derive both color and LBSP matching distance thresholds from a common value, $R_t(x)$. The adjustment control loop behind this parameter based on our two pixel-level indicators can be described as

$$R_t(x) = \begin{cases} R_{t-1}(x) + \lambda_R \cdot v_t(x) & \text{if } R_{t-1}(x) < (1 + \bar{d}_t(x) \cdot 2)^2 \\ R_{t-1}(x) - \dfrac{\lambda_R}{v_t(x)} & \text{otherwise} \end{cases} \tag{13}$$

where $\lambda_R \in [0, 1]$ is a fixed scaling factor. Note that $R_t(x)$ can only take values greater or equal to 1; this lower limit reflects the baseline matching distance threshold used in perfectly static regions. We control $R_t(x)$ variations via $\bar{d}_t(x)$ based on an exponential relation since it allows much easier feature matching in highly unstable regions (i.e. when $\bar{d}_t(x) \gg 0$). Here, $v_t(x)$ directly controls the variation step size of $R_t(x)$. In static regions, it prevents $R_t(x)$ from increasing too fast (which helps against camouflage problems), and in dynamic regions, it prevents it from decreasing too fast while $\bar{d}_t(x)$ fluctuates.

Feature matching distances and local persistence sums are closely related in the feedback process as both influence $\bar{d}_t(x)$ through (10). Therefore, we tie the adjustment mechanism of persistence thresholds to $R_t(x)$, and define it as

$$W_t(x) = \frac{q_t(\omega_1)}{R_{t-1}(x) \cdot 2} \tag{14}$$

where ω_1 is the first word of $B_l(x)$, and thus its most persistent one due to sorting. The idea here is to always have at least one local word with enough persistence to classify a pixel as background. This also means that the value of $W_t(x)$ is kept in the persistence range dictated by the words of $B_l(x)$, and each pixel model can have a unique classification behavior.

While our pixel-level controllers rely on segmentation noise to provide rapid parameter adjustments, this type of noise is an inherent characteristic of pixel-based segmentation due to shot noise in the analyzed images. Segmentation noise can also be easily eliminated afterwards using post-processing or regularization techniques. In PAWCS, we ultimately clean the raw segmentation maps S_t by using median blurring and morphological operations. This essentially mimics the effect of a more complex frame-level regularization approach at a low cost.

D. Model adaptability

The pixel-level controllers are mostly responsible for the overall flexibility of our method, but we also define three heuristics to further improve model adaptability. The first one relies on texture analysis to provoke extra updates in uniform background regions. This leads to stronger word persistence in local dictionaries, and thus better long-term word retention. In short, we cut down the value of $T_t(x)$ when the observed LBSP descriptor in $I_t(x)$ is “flat” (i.e. its Hamming weight is close to zero), which causes update mechanisms to trigger more often. The assumption here is that uniform regions are better background candidates than regions with strong gradients. Cluttered regions are largely unaffected by this criterion; this indirectly helps prevent “ghosts” from forming in highly textured background regions due to small, temporary texture displacements.

The second heuristic we use is based on segmentation instability. Large, long-term discrepancies between the “raw” output segmentation $S_t(x)$ and its post-processed equivalent are not as likely due to alternating labels than to small dynamic background regions (e.g. leaves in a tree). Our local feature description approach is very sensitive to such small variations; to help post-processing, we double the fixed R_d offset in (3b) when discrepancies above a fixed threshold are detected for x. This reduces the texture change detection sensitivity in the matching process and eliminates more false positive classifications in those regions.

Our last heuristic is responsible for spreading illumination updates between neighboring pixel models and causing chain reactions that allow large background surfaces to be updated rapidly. As stated in Section III-A, the color component of words can sometimes be updated to account for gradual changes in the background. To propagate these updates, we keep a 2D map of where they happen; then, for a pixel x, if one of its neighbors was recently updated, we halve the value of $T_t(x)$. This approach allows our model to respond rapidly and efficiently to frame-wide lighting variations.

IV. Experiments

The state-of-the-art presented in Section II makes it clear that traditional datasets (e.g. [36], [56]) are too small and no longer challenging to modern background subtraction methods. Moreover, classic methods (e.g. [15], [16]) have long since been surpassed and no longer offer a good performance reference. To properly evaluate our method, we rely on the 2012 and 2014 versions of the ChangeDetection.net (CDnet) benchmark and dataset [1], [11], [12]. Unlike older alternatives, the CDnet dataset offers a wide variety of real-world sequences split into eleven categories based on the challenges they contain. Totaling nearly 160,000 manually annotated frames, it is several orders of magnitude larger than other real-world datasets, and multiple times larger than those based on synthetic data. Part of the groundtruth is also withheld and kept for online testing to prevent overtuning.

In Section IV-A and IV-B, we respectively discuss our 2012 and 2014 CDnet results, and compare them to those of 27 methods listed online[2] or self-reported in prior publications (we report only the top performers). Note that not all methods tested on the 2012 dataset have been tested on the 2014 version, as the latter is much harder, and some authors prefer focusing on a smaller subset of segmentation challenges. Finally, we discuss memory footprint, processing speed and the possibility of a parallel implementation in Section IV-C.

We use the same PAWCS configuration to process all sequences of the 2012 and 2014 versions of the CDnet dataset. This is a disadvantage, as methods which were only tested on the 2012 version were specifically tuned for only those categories. We assume all self-reported results used for comparisons followed the tuning guidelines of the CDnet benchmark. To determine which parameters to use for LBSP feature description, we followed the approach of [8], which dynamically balances them based on the observed scene's gradient content. We also used the frame-level component presented in [8] to detect drastic changes (e.g. light switch events) in the analyzed sequences, and automatically reset our model when needed. Besides, we fix the value of the metaparameters presented in the previous sections as follows:

- Word weight offset value: $t_0 = 1000$
- Maximum number of words per dictionary: $N = 50$
- Baseline color distance threshold: $R_c = 20$
- Baseline LBSP distance threshold: $R_d = 2$
- Feedback recurrence adaptation rate: $\alpha = 0.01$
- Local update rate change factor: $\lambda_T = 0.5$
- Local distance threshold change factor: $\lambda_R = 0.01$

For more details on the post-processing steps and feedback heuristics of PAWCS, the reader is invited to refer to our implementation[3].

We limit the presentation of qualitative results (Figures 4 and 5) due to space constraints and because the difference between state-of-the-art methods and groundtruth is sometimes hard to perceive. Our full segmentation results can be downloaded via the CDnet evaluation platform, where more granular comparisons are also possible with most methods based on seven evaluation metrics.

TABLE I
Segmentation results of PAWCS on CDnet2012

Category	Re	Pr	FM	MCC
baseline	0.941	0.939	0.940	0.938
camera jitter	0.784	0.866	0.814	0.812
dynamic background	0.887	0.904	0.894	0.894
interm. obj. motion	0.749	0.839	0.776	0.774
shadow	0.917	0.871	0.891	0.887
thermal	0.850	0.828	0.832	0.828
overall	0.855	0.875	0.858	0.855

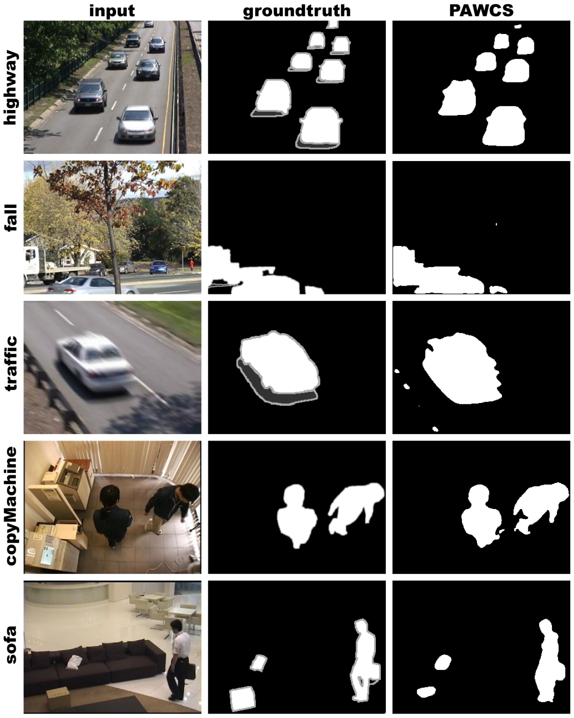


Fig. 4. Qualitative comparison of our segmentation results with the groundtruth on various sequences of CDnet2012. Gray regions in the groundtruth are not evaluated.

A. CDnet 2012

We first present in Table I the average Recall (Re), Precision (Pr), F-Measure (FM) and Matthew's Correlation Coefficient (MCC) scores of PAWCS on CDnet2012. The definition of the first three metrics can be found in [11]. MCC is also used here since it provides a good assessment of overall performance in unbalanced binary classification problems, which background subtraction falls into. It is defined by

$$\mathrm{MCC} = \frac{(\mathrm{TP}\cdot\mathrm{TN}) - (\mathrm{FP}\cdot\mathrm{FN})}{\sqrt{(\mathrm{TP{+}FP})\cdot(\mathrm{TP{+}FN})\cdot(\mathrm{TN{+}FP})\cdot(\mathrm{TN{+}FN})}}, \quad (15)$$

where True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) are defined like in [11].

Note that the description of each challenge, category and sequence is provided in [1], [11]. We can see from Table I

[2] http://www.changedetection.net

[3] https://github.com/plstcharles/litiv

TABLE II
AVERAGE PER-CATEGORY AND OVERALL SCORES ON CDNET2012 [a]

Method	Baseline	Cam. Jitt.	Dyn. Bg	Int. Mot.	Shadow	Thermal	Overall (2012)		
	FM	FM	FM	FM	FM	FM	Re	Pr	FM↓
PAWCS (proposed)	*0.940*	0.814	**0.894**	0.776	*0.891*	**0.832**	**0.855**	**0.875**	**0.858**
Shared GMM [57]†*	0.935	*0.817*	0.867	**0.798**	0.813	*0.825*	-	-	*0.842*
FTSG [38]*	0.933	0.751	*0.879*	*0.789*	0.883	0.777	*0.838*	*0.867*	0.835
SuBSENSE [8]*	**0.950**	0.815	0.818	0.657	**0.899**	0.817	0.828	0.858	0.826
MBS V0 [58]*	0.928	**0.836**	0.790	0.709	0.778	0.811	0.790	0.849	0.809
2-pass RPCA [31]†	0.928	0.815	0.782	0.653	0.806	0.760	0.799	0.798	0.796
AMBER+ [50]*	0.881	0.711	0.843	0.721	0.813	0.760	0.791	0.827	0.788
CwisarD [59]	0.908	0.781	0.809	0.567	0.841	0.762	0.818	0.774	0.778
Spectral-360 [60]	0.933	0.716	0.787	0.566	0.884	0.776	0.777	0.846	0.777
DPGMM [18]	0.929	0.748	0.814	0.542	0.813	0.813	0.827	0.793	0.776
BMTDL [39]†	0.877	0.725	0.753	0.686	0.810	0.793	0.785	0.773	0.774
M.-H. Yang's [24]†	0.883	0.787	0.808	0.525	0.860	0.754	0.828	0.786	0.769
ST-HBF [40]†	0.934	0.712	0.828	0.535	0.864	0.735	-	-	0.768 [b]
SGMM-SOD [61]	0.921	0.672	0.688	0.715	0.865	0.735	0.768	0.835	0.766
DECOLOR [30], [31]	0.923	0.778	0.708	0.595	0.832	0.708	0.801	0.727	0.757
PBAS [10]	0.924	0.722	0.683	0.575	0.860	0.756	0.784	0.816	0.753
EFIC [62]*	0.917	0.713	0.578	0.578	0.820	0.838	0.809	0.741	0.741
PSP-MRF [47]	0.929	0.750	0.696	0.565	0.791	0.693	0.804	0.751	0.737
PCP [29], [31]	0.911	0.722	0.694	0.537	0.789	0.719	0.701	0.776	0.729
SC-SOBS [41]	0.933	0.705	0.669	0.592	0.779	0.692	0.802	0.732	0.728
CDPS [63]	0.921	0.487	0.750	0.741	0.809	0.662	0.777	0.761	0.728
AAPSA [43]*	0.918	0.721	0.671	0.510	0.795	0.703	0.708	0.801	0.720
SMSOM-BM [42]†	0.927	0.634	0.675	-	-	0.793	-	-	-

[a] Red-bold entries indicate the best result in a given column, and blue-italics the second best.
[b] We recalculated the overall result of [40] based on the CDnet evaluation guidelines.
* Extracted from CDnet2014 results.
† Self-reported.

that our method offers good balance between precision and recall in all test categories but "camera jitter" and "intermittent object motion". In the former, vibrating cameras cause the entire observed scenes to be perceived as dynamic. This means that pixel model updates trigger very often and that static foreground objects become part of them very fast, which leads to lower Recall scores. The second problematic category is of particular interest here: "intermittent object motion" contains four videos focused on long-term immobile object segmentation, and two on "ghost" elimination (i.e. purging the model of background objects removed from the scene). A lower Recall score in this category indicates that some foreground objects were eventually "lost" to the background model over time. In reality, this difference is not surprising, as very little time is allowed to "learn" the empty background, therefore our words never attain "strong" persistence values. This category is also by far the hardest in CDnet 2012; we discuss it in detail later in this section. The other categories shown in Table I deal with more traditional challenges of background subtraction, and our method performs well in all of them.

Next, we present in Table II a compilation of the per-category and overall F-Measure scores of the 23 top-performing methods published and available online as of July 2015. Like [8], [18], we do not use the official CDnet rankings in these results, as adding and removing methods from the comparison pool (no matter their performance) can drastically affect their final ordering. The F-Measure scores used instead were found to be closely correlated with the method rankings in all three previous CDnet evaluation reports [1], [11], [12]. F-Measure is also the most common metric used for comparison in the literature, and thus allows us to list practically all self-reported results of methods not yet on the online benchmark.

Table II shows that PAWCS outperforms all other methods based on overall Recall, Precision, and F-Measure scores for CDnet2012. It also places first or second for F-Measure in four out of six categories, with a noticeable gap to the state-of-the-art only in "intermittent object motion". Even in the "camera jitter" category, the results of PAWCS are only slightly worse than those of the best methods, which rely on multiple configurations [57], prior image realignment [31], or supervised training [58] to address the shaking camera challenge. This means our dictionary update and feedback mechanisms are already quite sufficient for this task. Besides, compared to others, PAWCS excels in the "shadow" category, offering performance similar to the performance of multiple methods in the less-challenging "baseline" category. In fact, as stated in [12], while "hard" shadows are still a challenge for all methods, soft shadows are no longer problematic to modern solutions. In our case, this is due to our illumination update mechanisms as well as our choice of features for word description. We can also note that overall, our "online" method outperforms all recent robust PCA-based methods [29]–[31] despite the fact that we "stream" the data in an online fashion (as opposed to offline batch-processing and optimization).

The advantages of our novel word consensus modeling strategy can be easily outlined by comparing our results to those of SuBSENSE [8] since both methods use similar features and feedback mechanisms. While the results of PAWCS are slightly worse in the "baseline" category and equivalent in "camera jitter" and "shadow", they are clearly superior in "dynamic background" and "intermittent object motion". This means our dictionaries can better reflect the multimodality of dynamic regions while staying sensitive to outliers, and

TABLE III
AVERAGE RECALL, PRECISION AND F-MEASURE SCORES ON THE INTERMITTENT OBJECT MOTION CATEGORY OF CDNET2012 [a]

Method	Recall	Precision	F-Measure↓
Shared GMM [57]†*	-	-	**0.798**
FTSG [38]*	**0.835**	**0.933**	*0.789*
PAWCS (proposed)	0.749	*0.839*	0.776
CDPS [63]	*0.808*	0.762	0.741
AMBER+ [50]*	0.762	0.753	0.721
SGMM-SOD [61]	0.736	0.814	0.715
MBS V0 [58]*	0.639	0.820	0.709
BMTDL [39]†	0.684	0.709	0.686
2-pass RPCA [31]†	0.653	0.715	0.683
SuBSENSE [8]*	0.658	0.796	0.657
SC-SOBS [41]	0.724	0.590	0.592
EFIC [62]*	0.742	0.563	0.578
PBAS [10]	0.670	0.705	0.575
Spectral-360 [60]	0.595	0.719	0.566
DPGMM [18]	0.676	0.653	0.542

[a] Red-bold entries indicate the best result in a given column, and blue-italics the second best.
* Extracted from CDnet2014 results.
† Self-reported.

TABLE IV
SEGMENTATION RESULTS OF PAWCS ON CDNET2014

Category	Re	Pr	FM	MCC [a]
bad weather	0.718	0.947	0.815	0.812
low framerate	0.773	0.641	0.659	0.657
night videos	0.361	0.654	0.415	0.432
pan-tilt-zoom	0.698	0.473	0.462	0.491
turbulence	0.812	0.681	0.645	0.786
overall (2014 only)	0.672	0.679	0.600	0.636
overall (2012+2014)	0.772	0.786	0.740	0.756

[a] Approximated based on available groundtruth.

that persistence is an ideal way to determine which words are more important for modeling based on past observations. The difference in the "baseline" category results can be explained by a slight increase in false negative classifications due to the global dictionary acting as a "fallback" model to prevent false positives elsewhere. Comparisons of some of our segmentation results with the groundtruth are shown in Figure 4.

Finally, we compare the results of various methods for the "intermittent object motion" category in Table III. We can note that most recent methods perform poorly when static foreground object and background ghosts are involved. The top performers, FTSG [38] and Shared-GMM [57], both rely on foreground modeling as well as region-level or object-level processing steps and heuristics to specifically tackle this category. On the other hand, our modeling approach only relies on the analysis of word persistence based on the principles detailed in Section III. Thus, we keep to low-level mechanisms without involving object shape semantics. Our design is therefore in accordance with the first Wallflower principle [36], but at a disadvantage when faced with these challenges.

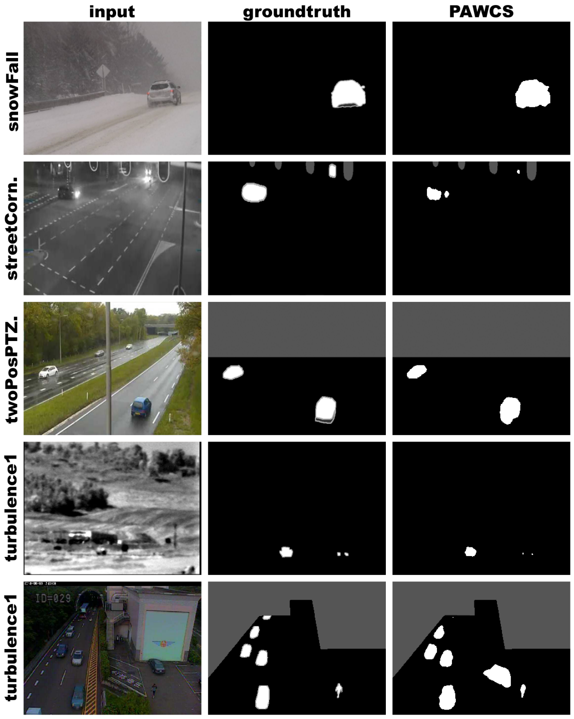


Fig. 5. Qualitative comparison of our segmentation results with the groundtruth on various sequences of CDnet2014. Gray regions in the groundtruth are not evaluated. An obvious false positive blob is visible in the last row's segmentation map; this is a temporary "ghost" artifact caused by a van that left the scene after being parked there.

B. CDnet 2014

As stated earlier, the 2014 dataset is much more complex than the original version. Its categories include videos captured outside during snowstorms, extremely low framerate videos with rapid illumination variations and color profile changes, highway surveillance videos captured at night with intense glare effects from car headlights, videos captured by pan-tilt-zoom (PTZ) cameras being operated, and thermal imaging videos of long-range surveillance in high temperature environments. Just like for the CDnet2012 dataset, we first present in Table IV PAWCS's performance on CDnet2014 using average Recall (Re), Precision (Pr), F-Measure (FM) and Matthew's Correlation Coefficient (MCC) scores. Note that since MCC is not computed by the online CDnet benchmark platform, and since half of the groundtruth is withheld, the MCC scores reported here are only representative of half of the dataset.

In comparison with the metrics shown for CDnet2012, it is quite clear that the 2014 dataset is more challenging. Only the "bad weather" category has F-Measure and MCC scores above 80%, and two categories have Precision or Recall scores below 50%. These Precision and Recall scores respectively indicate that more than half of foreground classifications were false, and more than half of all true foreground classifications were missed by our method. Surprisingly, PAWCS performed better in the "pan-tilt-zoom" category (in which the basic assumption of the static camera is violated) than in "night videos". This can be explained by two factors: first, our global dictionary allows large uniform regions (which make up the bulk of all background areas in PTZ videos) to be properly recognized as background despite important camera motion. Second, due

TABLE V
AVERAGE PER-CATEGORY AND OVERALL SCORES ON CDNET2014 [a]

Method	Bad Weath.	Low Fr.	Night Vid.	PTZ	Turbul.	Overall (2014)			Overall (2012+2014)		
	FM	FM	FM	FM	FM	Re	Pr	FM	Re	Pr	FM↓
SuBSENSE [8]	**0.862**	0.645	*0.560*	0.348	**0.779**	**0.794**	0.623	*0.639*	**0.812**	0.751	**0.741**
PAWCS (proposed)	0.815	*0.659*	0.415	0.462	0.645	0.672	*0.679*	0.600	0.772	**0.786**	*0.740*
FTSG [38]	*0.823*	0.626	0.513	0.324	0.713	0.678	0.652	0.600	0.766	0.770	0.728
MBS V0 [58]	0.773	0.628	0.516	*0.512*	0.570	0.635	0.617	0.599	0.719	0.744	0.714
EFIC [62]	0.779	**0.663**	**0.655**	**0.584**	0.671	*0.758*	**0.701**	**0.670**	*0.786*	0.722	0.709
CwisarDH [44]	0.684	0.641	0.374	0.322	0.723	0.531	0.678	0.549	0.661	*0.773*	0.681
Spectral-360 [60]	0.757	0.644	0.483	0.365	0.543	0.693	0.540	0.558	0.735	0.705	0.673
AMBER+ [50]	0.767	0.469	0.380	0.135	*0.755*	0.599	0.584	0.501	0.704	0.716	0.658
AAPSA [43]	0.774	0.494	0.416	0.330	0.464	0.579	0.561	0.496	0.650	0.692	0.618
SC-SOBS [41]	0.662	0.546	0.450	0.041	0.488	0.715	0.462	0.437	0.762	0.609	0.596
KNN [9]	0.759	0.549	0.420	0.213	0.520	0.658	0.547	0.492	0.665	0.679	0.594
RMoG [64]	0.683	0.531	0.427	0.247	0.458	0.582	0.543	0.469	0.594	0.697	0.574
GMM [15]	0.738	0.537	0.410	0.152	0.466	0.653	0.484	0.461	0.685	0.603	0.571
KDE [16]	0.757	0.548	0.436	0.037	0.448	0.729	0.457	0.445	0.738	0.581	0.569

[a] Red-bold entries indicate the best result in a given column, and blue-italics the second best.

to the use of LBSP descriptors, and since our dictionaries are kept at minimal word counts, our method is sensitive to noise and color variations in low contrast background regions (such as the ones in "night videos"). On the other hand, the "turbulence" and "bad weather" categories also mostly contain low contrast videos with dynamic background regions, making it a combination of very challenging problems. Finally, the "low framerate" category shows decent results in all but one sequence (not shown here), filmed at one frame per six seconds. This problematic sequence presents very large color variations along with dynamic background elements, making it very hard to process.

We compare in Table V the scores obtained by 14 methods which were tested thus far on CDnet2014. We omit the 2014 dataset results of [57] as their published results are significantly different from those reported online. Again, the performance of PAWCS is well above the average, ranking second in overall F-Measure (by a marginal difference) to [8], and third in overall F-Measure for the 2014 categories only. On the other hand, on the official CDnet rankings available online based on all seven evaluation metrics (not shown here) list PAWCS as the best method by a good margin.

The F-Measure scores of PAWCS are lower than those of the state-of-the-art in the "bad weather", "night videos" and "turbulence" categories. As mentioned before, this is due to the dynamic background/low contrast combination of challenges present in those sequences, to which our low-complexity modeling approach has difficulty adapting. Still, the overall 2012+2014 F-Measure score of PAWCS demonstrates that it is very flexible, and that it tackles most challenges without compromising too much of its performance elsewhere. In terms of Recall and Precision, we can observe balanced scores that are higher than those of most methods in both dataset versions, further demonstrating the overall flexibility of our approach. We present some qualitative comparisons between the groundtruth and our segmentation results in Figure 5.

C. Processing speed and memory footprint

Our C++ implementation processes the entire CDnet2012 dataset on a third generation, quad-core Intel i5 CPU (one sequence per core) at 22 frames per second, and it processes individual QVGA sequences on a single core at 15 frames per second. This is about 50% slower than [8] given equal model sizes for both methods, but still much faster than most video segmentation methods. Comparing this result to others is difficult due to the lack of open-source implementations available online; we offer ours for future reference.

Out of the entire processing load for a single frame, about 75% of it is for parsing dictionaries for matches and updating them, 15% is for feedback mechanisms, 5% is for frame-wide operations on global word persistence maps, and 5% is for output regularization via morphological operations and median blurring. The complexity of PAWCS is constant with respect to the number of pixels in the input frames. Given the ample opportunities for parallelization, real-time processing of very high resolution videos appears achievable. We recently implemented a similar non-parametric sample consensus method [23] on GPU, and reached a speedup of 30x over its CPU implementation (i.e. it could process several thousand frames per second). This speed was capped only by the memory bandwidth of the hardware we used.

As for the memory footprint of our proposed method, note that a background word requires three bytes of memory per channel to store color information and LBSP binary strings (based on the 5x5 pattern of [55]), and three integers to store its persistence parameters (four bytes each, if $t_{\max} > 2^{16}$ is expected). In a worse-case scenario where all pixel models require $N = 50$ background words (and those words are never discarded), and where the target video is RGB 1080p ($\approx$2.1 megapixels), the memory requirement set by our pixel modeling approach would be slightly over 2 GB. Global words would also add less than 2 MB each to the total, given that their persistence maps are implemented using one byte per pixel. For embedded/mobile applications, given an average of $N = 5$ active words per pixel model and a QVGA resolution, the memory requirement of PAWCS would be less than 10 MB.

V. CONCLUSION

"Word consensus", a new non-parametric pixel-level modeling approach, has been presented. It works by capturing local image samples and evaluating their recurrence among recent observations. We showed through our experiments with

PAWCS that word consensus is performant when tackling segmentation challenges involving static foreground objects and multimodal background regions. Its ability to automatically deduce which model samples (or words) are the most important background components based on temporal persistence allows it to keep a low overall memory footprint. With the addition of closed-loop controllers and other feedback mechanisms, our complete method allows each targeted image pixel to behave differently in terms of classification behavior and model complexity. A frame-level word dictionary is also considered to increase spatial coherence between pixel models and prevent false classifications due to large-scale background change patterns.

Our results showed that our method is superior to most in scenarios with traditional and modern background subtraction challenges. There is still room for improvement however; using a more sophisticated output regularization step (e.g. [47]), or explicitly modeling the foreground appearance of objects pixel-wise are good avenues for future work.

References

[1] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "A novel video dataset for change detection benchmarking," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4663–4679, 2014.

[2] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer Science Review*, vol. 1112, pp. 31–66, 2014.

[3] S. Brutzer, B. Hoferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2011, pp. 1937–1944.

[4] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1337–1342, Oct 2003.

[5] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, Jun. 2005.

[6] 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, 2011.

[7] M. Van Droogenbroeck and O. Barnich, "ViBe: A disruptive method for background subtraction," in *Background Modeling and Foreground Detection for Video Surveillance*, T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, Eds. Chapman and Hall/CRC, June 2014, ch. 7.

[8] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 359–373, 2015.

[9] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, 2006.

[10] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 38–43.

[11] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2012, pp. 1–8.

[12] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2014, pp. 387–394.

[13] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *Proc. IEEE Winter Conf. Applicat. Comput. Vis.*, Jan 2015, pp. 990–997.

[14] M. Van Droogenbroeck and O. Paquot, "Background subtraction: Experiments and improvements for ViBe," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 32–37.

[15] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 1999, pp. 246–252.

[16] A. M. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric model for background subtraction," in *Proc. 6th European Conf. Comput. Vis.*, 2000, pp. 751–767.

[17] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-Based Surveillance Systems*. Springer US, 2002, pp. 135–144.

[18] T. Haines and T. Xiang, "Background subtraction with dirichlet process mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 670–683, April 2014.

[19] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. S. Davis, "Background modeling and subtraction by codebook construction," in *Proc. IEEE Int. Conf. Image Process.*, 2004, pp. 3061–3064.

[20] M. Wu and X. Peng, "Spatio-temporal context for codebook-based dynamic background subtraction," *Int. J. Electron. Commun.*, vol. 64, no. 8, pp. 739–747, 2010.

[21] B. Mayer and J. Mundy, "Duration dependent codebooks for change detection," in *Proc. of the British Machine Vis. Conf.*, 2014.

[22] 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.

[23] P.-L. St-Charles and G.-A. Bilodeau, "Improving background subtraction using local binary similarity patterns," in *Proc. IEEE Winter Conf. Applicat. Comput. Vis.*, March 2014, pp. 509–515.

[24] M.-H. Yang, C.-R. Huang, W.-C. Liu, S.-Z. Lin, and K.-T. Chuang, "Binary descriptor based nonparametric background modeling for foreground extraction by using detection theory," vol. 25, no. 4, pp. 595–608, 2015.

[25] C. Silva, T. Bouwmans, and C. Frelicot, "An eXtended center-symmetric local binary pattern for background modeling and subtraction in videos," in *Proc. Int. Conf. Comput. Vis. Theory and Appl.*, 2015, pp. 395–402.

[26] L. Lin, Y. Xu, X. Liang, and J. Lai, "Complex background subtraction by pursuing dynamic spatio-temporal models," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 3191–3202, 2014.

[27] M. Braham and M. Van Droogenbroeck, "A generic feature selection method for background subtraction using global foreground models," in *Proc. Advanced Concepts for Intell. Vis. Syst.*, 2015. [Online]. Available: http://hdl.handle.net/2268/185047

[28] N. Oliver, B. Rosario, and A. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, Aug 2000.

[29] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11–37, 2011.

[30] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, 2013.

[31] Z. Gao, L.-F. Cheong, and Y.-X. Wang, "Block-sparse RPCA for salient motion detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 1975–1987, 2014.

[32] Y. Sun, X. Tao, Y. Li, and J. Lu, "Robust 2D principal component analysis: A structured sparsity regularized approach," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2515–2526, 2015.

[33] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1568–1575.

[34] J. Feng, H. Xu, and S. Yan, "Online robust PCA via stochastic optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 404–412.

[35] F. Seidel, C. Hage, and M. Kleinsteuber, "pROST: A smoothed l_p-norm robust online subspace tracking method for background subtraction in video," *Mach. Vis. Appl.*, vol. 25, no. 5, pp. 1227–1240, 2014.

[36] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1, 1999, pp. 255–261 vol.1.

[37] Y.-T. Chen, C.-S. Chen, C.-R. Huang, and Y.-P. Hung, "Efficient hierarchical method for background subtraction," *Pattern Recognit.*, vol. 40, no. 10, pp. 2706–2715, 2007.

[38] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Static and moving object detection using flux tensor with split gaussian models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2014.

[39] A. Stagliano, N. Noceti, A. Verri, and F. Odone, "Online space-variant background modeling with sparse coding," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2415–2428, 2015.

[40] B. Dey and M. Kundu, "Efficient foreground extraction from HEVC compressed video for application to real-time analysis of surveillance 'big' data," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3574–3585, 2015.

[41] L. Maddalena and A. Petrosino, "The SOBS algorithm: What are the limits?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 21–26.

[42] Z. Zhao, X. Zhang, and Y. Fang, "Stacked multilayer self-organizing map for background modeling," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2841–2850, Sept 2015.

[43] G. Ramírez-Alonso and M. I. Chacón-Murguía, "Auto-adaptive parallel SOM architecture with a modular analysis for dynamic object segmentation in videos," *Neurocomputing*, vol. 175B, pp. 990–1000, 2016.

[44] M. D. Gregorio and M. Giordano, "Change detection with weightless neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2014.

[45] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *Proc. Int. Conf. Systems, Signals and Image Proc.*, 2016. [Online]. Available: http://hdl.handle.net/2268/195180

[46] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov 2005.

[47] A. Schick, M. Bauml, and R. Stiefelhagen, "Improving foreground segmentations with probabilistic superpixel markov random fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 27–31.

[48] H.-H. Lin, J.-H. Chuang, and T.-L. Liu, "Regularized background adaptation: A novel learning rate control scheme for gaussian mixture modeling," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 822–836, 2011.

[49] Z. Chen and T. Ellis, "A self-adaptive gaussian mixture model," *Comput. Vis. and Image Understanding*, vol. 122, pp. 35–46, 2014.

[50] B. Wang and P. Dudek, "A fast self-tuning background subtraction algorithm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2014.

[51] S. Bianco, G. Ciocca, and R. Schettini, "How far can you get by combining change detection algorithms?" *CoRR*, vol. abs/1505.02921, 2015. [Online]. Available: http://arxiv.org/abs/1505.02921

[52] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. and Image Understanding*, vol. 122, pp. 4–21, 2014.

[53] T. Bouwmans and E. H. Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *Comput. Vis. and Image Understanding*, vol. 122, pp. 22–34, 2014.

[54] H. Wang and D. Suter, "Background subtraction based on a robust consensus method," in *Proc. IEEE Int. Conf. Pattern Recognit.*, vol. 1, 2006, pp. 223–226.

[55] G.-A. Bilodeau, J.-P. Jodoin, and N. Saunier, "Change detection in feature space using local binary similarity patterns," in *Int. Conf. Comput. Robot Vis.*, 2013, pp. 106–112.

[56] L. Li, W. Huang, I.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov 2004.

[57] Y. Chen, J. Wang, and H. Lu, "Learning sharable models for robust background subtraction," in *Proc. IEEE Int. Conf. Multimultimedia and Expo*, 2015, pp. 1–6.

[58] H. Sajid and S.-C. S. Cheung, "Background subtraction for static and moving camera," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 4530–4534.

[59] M. D. Gregorio and M. Giordano, "A WiSARD-based approach to CDnet," in *Proc. 1st BRICS Countries Congress*, 2013.

[60] M. Sedky, M. Moniri, and C. C. Chibelushi, "Spectral-360: A physics-based technique for change detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2014.

[61] R. Evangelio, M. Patzold, I. Keller, and T. Sikora, "Adaptively splitted GMM with feedback improvement for the task of background subtraction," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 5, pp. 863–874, 2014.

[62] G. Allebosch, F. Deboeverie, P. Veelaert, and W. Philips, "Efic: Edge based foreground background segmentation and interior classification for dynamic camera viewpoints," in *Proc. Advanced Concepts for Intell. Vis. Syst.*, 2015, pp. 130–141.

[63] F. Hernandez-Lopez and M. Rivera, "Change detection by probabilistic segmentation from monocular view," *Mach. Vis. Appl.*, vol. 25, no. 5, pp. 1175–1195, 2014.

[64] S. Varadarajan, P. Miller, and H. Zhou, "Spatial mixture of gaussians for dynamic background modelling," in *Proc. IEEE Int. Conf. Adv. Video and Signal Based Surveillance*, 2013, pp. 63–68.

Pierre-Luc St-Charles (S'13) received the B.Eng. degree in computer engineering from Polytechnique Montréal, QC, Canada, in 2013. He is currently enrolled in a Ph.D. program at the same institution, and has been working part-time at the Computer Research Institute of Montréal (CRIM) since 2011. His research interests include image and video segmentation, multimodal registration, and video surveillance applications.

Guillaume-Alexandre Bilodeau (M'10) received the B.Sc.A. degree in computer engineering and the Ph.D. degree in electrical engineering from Université Laval, QC, Canada, in 1997 and 2004, respectively.

In 2004, he was appointed Assistant Professor at Polytechnique Montréal, QC, Canada, and Associate Professor in 2011. Since 2014, he has been a Full Professor at the same institution. His research interests encompass image and video processing, video surveillance, object recognition, content-based image retrieval, and medical applications of computer vision.

Dr. Bilodeau is a member of the Province of Québec's Association of Professional Engineers (OIQ) and of the REPARTI research network.

Robert Bergevin (M'84) received the B.Eng. degree in Electrical Engineering and the M.A.Sc. degree in Biomedical Engineering from Polytechnique Montréal and the Ph.D. degree in Electrical Engineering from McGill University. His research interests are in image analysis and cognitive computer vision. His main works address generic modeling and recognition of objects in static images and tracking and modeling of people and animals in image sequences.

Dr. Bergevin is a member of the Computer Vision and Systems Laboratory at Laval University, Québec City, where he is a Professor in the Department of Electrical and Computer Engineering. He is a member of the Province of Québec's Association of Professional Engineers (OIQ) and serves as Area Editor for the Computer Vision and Image Understanding journal.