P.-M. Jodoin, S. Piérard, Y. Wang, and M. Van Droogenbroeck. **Background Modeling and Foreground Detection for Video Surveillance**. In *T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, editors, "Background Modeling and Foreground Detection for Video Surveillance",* chapter 24, CRC Press, July 2014.

1

Overview and Benchmarking of Motion Detection Methods

Pierre-Marc Jodoin
Université de Sherbrooke, Canada

Sébastien Piérard
Université de Liège, Belgium

Yi Wang
Université de Sherbrooke, Canada

Marc Van Droogenbroeck
Université de Liège, Belgium

1.1 Introduction

Motion detection is closely coupled with higher level inference tasks such as detection, localization, tracking, and classification of moving objects, and is often considered to be a preprocessing step. Its importance can be gauged by the large number of algorithms that have been developed to-date and the even larger number of articles that have been published on this topic. A quick search for 'motion detection' on IEEE Xplore© returns over 20,000 papers. This shows that motion detection is a fundamental topic for a wide range of video analytic applications. It also shows that the number of motion detection methods proposed so far is impressively large.

Among the many variants of motion detection algorithms, there seems to be no single algorithm that competently addresses all of the inherent real world challenges. Such challenges are sudden illumination variations, night scenes, background movements, illumination changes, low frame rate, shadows, camouflage effects (photometric similarity of object and background) and ghosting artifacts (delayed detection of a moving object after it has moved away) to name a few.

In this chapter, we provide an overview of the most highly cited motion detection methods. We identify the most commonly used background models together with their features, the kind of updating scheme they use, some spatial aggregation models as well as the most widely used post-processing operations. We also provide an overview of datasets used to validate motion detection methods. Please note that this literature review is by no means exhaustive and thus we provide a list of surveys that the reader can rely on for further details.

Together with this overview, we provide benchmarking results on different categories of videos, for different methods, different features and different post-processing methods. This should give the reader a perspective on some of the most effective methods available nowadays on different types of videos. We also report results on majority vote strategies that we use to combine the output of several methods. The goal being to identify complementary methods whose combination helps improving results. All benchmark results have been obtained on the changedetection.net dataset.

1.2 Motion Detection Methods

Motion detection is often achieved by building a representation of the scene, called background model, and then observing deviations from this model for each incoming frame. A sufficient change from the background model is assumed to indicate a moving object. In this document, we report on the most commonly-used models which we refer to as the *basic*, *parametric*, *non-parametric*, *data-driven* and *matrix decomposition* models. Other models for motion detection are also accounted for in this section such as the *prediction* model, the *motion segmentation* model, and the *machine learning* approaches. All these motion detection methods are summarized in Table 1.1.

Together with these 8 families of methods, we review commonly-used features, spatial aggregation techniques, updating scheme as well as post-processing methods.

1.2.1 Basic Models

The simplest strategy to detect motion is to subtract the pixel's color in the current frame from the corresponding pixel's color in the background model [14]. A temporal median filter can be used to estimate a color-based background model [110]. One can also generalize to other features such as color histograms [84, 196] and local self-similarity features [70]. In general, these temporal filtering methods are sensitive to compression artifacts, global illumination changes and incapable to detect moving objects once they become stationary.

Frame differencing is another basic model. It aims to detect changes in the state of a pixel by subtracting the pixel's intensity (or color) in the current frame from its intensity (or color) in the previous frame. Although this method is computationally inexpensive, it cannot detect a moving object once it stops moving or when the object motion becomes small; instead it typically detects object boundaries, covered and exposed areas due to object motion.

Motion history images [16, 118] are also used as a basic model for motion detection. It is obtained by successive layering of frame differences. For each new frame, existing frame differences are scaled down in amplitude, subject to some threshold, and the new motion label field is overlaid using its full amplitude range. In consequence, image dynamics ranging from two consecutive frames to several dozen frames can be captured in a single image.

1.2.2 Parametric Models

In order to improve robustness to noise, parasite artifacts, and background motion, the use of a per-pixel Gaussian model has been proposed [182]. In a first step, the mean and standard deviation are computed for each pixel. Then, for each frame, the likelihood of each pixel color is determined and pixels whose probability is below a certain threshold are labeled as foreground pixels. Since pixels in noisy areas are given a larger standard deviation, a larger color variation is needed in those areas to detect motion. This is a fundamental difference with the basic models for which the tolerance is fixed for every pixel. As shown by Kim *et al.* [80], a generalized Gaussian model can also be used and Morde *et al.* [118] have shown that a Chebychev inequality can also improve results. With this model, the detection criteria depends on how many standard deviations a color is from

P.-M. Jodoin, S. Piérard, Y. Wang, and M. Van Droogenbroeck. **Background Modeling and Foreground Detection for Video Surveillance**. In *T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, editors, "Background Modeling and Foreground Detection for Video Surveillance",* chapter 24, CRC Press, July 2014.

the mean.

A single Gaussian, however, is not a good model for dynamic scenes [52] as multiple colors may be observed at a pixel due to repetitive object motion, shadows or reflectance changes. A substantial improvement is achieved by using multiple statistical models to describe background color. A Gaussian Mixture Model (GMM) [156] was proposed to represent each background pixel. GMM compares each pixel in the current frame with every model in the mixture until a matching Gaussian is found. If a match is found, the mean and variance of the matching Gaussian are updated, otherwise a new Gaussian with the mean equal to the current pixel color and some initial variance is introduced into the mixture. Instead of relying on only one pixel, GMM can be trained to incorporate extended spatial information [72].

Several papers [74] improved the GMM approach to add robustness when shadows are present and to make the background models more adaptive to parasitic background motion. A recursive method with an improved update of the Gaussian parameters and an automatic selection of the number of modes was presented in [202]. Haines *et al.* [58] also propose an automatic mode selection method, but with a Dirichlet process. A splitting GMM that relies on a new initialization procedure and a mode splitting rule was proposed in [46, 48] to avoid over-dominating modes and resolve problems due to newly static objects and moved away background objects while a multi-resolution block-based version was introduced in [146]. The GMM approach can also be expanded to include the generalized Gaussian model [4].

As an alternative to mixture models, Bayesian approaches have been proposed. In [138], each pixel is modeled as a combination of layered Gaussians. Recursive Bayesian update instead of the conventional expectation maximization fitting is performed to update the background parameters and better preserve the multi-modality of the background model. A similar Bayesian decision rule with various features and a learning method that adapt to both sudden and gradual illumination changes in used in [94].

Another alternative to GMM is background clustering. In this case, each background pixel is assigned a certain number of clusters depending on the color variation observed in the training video sequence. Then, each incoming pixel whose color is close to a background cluster is considered part of the background. The clustering can be done using K-means (or a variant of it) [26, 126] or codebook [82].

1.2.3 Non-Parametric Methods

In contrast to parametric models, non-parametric Kernel Density Estimation (KDE) fits a smooth probability density function to a time window with previously-observed pixel values at the same location [42]. During the change detection process, a new-frame pixel is tested against its own density function as well as those of pixels nearby. This increases the robustness against camera jitter or small movements in the background. Similar effects can be achieved by extending the support to larger blocks and using texture features that are less sensitive to inter-frame illumination variations. Mittal and Poggio [116] have shown that robustness to background motion can be increased by using variable-bandwidth kernels.

Although nonparametric models are robust against small changes, they are expensive both computationally and in terms of memory use. Moreover, extending the support causes small foreground objects to disappear. As a consequence, several authors worked to improve the KDE model. For instance, a multi-level method [122] makes KDE computationally independent of the number of samples. A trend feature can also be used to reliably differentiate periodic background motion from illumination changes [190].

1.2.4 Data-driven Methods

Recently, pixel-based data-driven methods using random samples for background modeling have shown robustness to several types of error sources. For example, ViBe [6, 170] not

P.-M. Jodoin, S. Piérard, Y. Wang, and M. Van Droogenbroeck. **Background Modeling and Foreground Detection for Video Surveillance**. In *T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, editors, "Background Modeling and Foreground Detection for Video Surveillance",* chapter 24, CRC Press, July 2014.

Background model	**References**
Basic	Running average [14, 84, 196, 70] Temporal median [110] Motion history image [16, 118]
Parametric	Single Gaussian [182] Gaussian Mixture Model (GMM) [156, 72, 74, 202, 48, 46, 146, 58] Background clustering [26, 126, 82] Generalized Gaussian Model [4, 80] Bayesian [138, 94] Chebyshev inequality [118]
None-Parametric	Kernel Density Estimation (KDE) [42, 122, 190, 116]
Data-driven	Cyclostationary [140] Stochastic K-nearest neighbors (KNN) [6, 64]. Deterministic KNN [202] Hidden Markov Model (HMM) [158]
Matrix Decomposition	Principal Component Analysis (PCA) [124, 188, 96, 36, 150] Sparsity and dictionary learning [136, 194]
Prediction Model	Kalman filter [78, 198] Weiner filter [168]
Motion Segmentation	Optical flow segmentation [180, 114, 102] GMM and Optical flow segmentation [126, 200]
Machine Learning	1-Class SVM [32] SVM [100, 62, 60] Neural networks [104, 106, 152]

TABLE 1.1 Overview of 8 families of motion detection methods.

only shows robustness to background motion and camera jitter but also to ghosting artifacts. Hofmann [64] improved the robustness of ViBe on a variety of difficult scenarios by automatically tuning its decision threshold and learning rate based on previous decisions made by the system. In both [6, 64], a pixel is declared as foreground if it is not close to a sufficient number of background samples from the past. A deterministic K nearest neighbor approach has also been proposed by Zivkovic and van der Heijiden [202], and one for non-parametric methods by Manzanera [108].

A shortcoming of the above methods is that they do not account for any "temporal correlation" within video sequences, thus they are sensitive to periodic (or near-periodic) background motion. For example, alternating light signals at an intersection, a flashing adversisement board, the appearance of rotating objects, etc. A cyclostationary background generation method based on frequency decomposition that explicitly harnesses the scene dynamics is proposed in [140]. In order to capture the cyclostationary behavior at each pixel, spectral coefficients of temporal intensity profiles are computed in temporal windows and a background model that is composed of those coefficients is maintained and fused with distance maps to eliminate trail effects.

An alternative approach is to use a Hidden Markov Model (HMM) with discrete states to model the intensity variations of a pixel in an image sequence. State transitions can then be used to detect changes [158]. The advantage of using HMMs is that certain events, which may not be modeled correctly by unsupervised algorithms, can be learned using the provided training samples.

1.2.5 Matrix Decomposition

Instead of modeling the variation of individual pixels, the whole image can be vectorized and used in background modeling. In [124], a holistic approach using eigenspace decomposition is proposed. For a certain number of input frames, a background matrix (called *eigen background*) is formed by arranging the vectorized representations of images in a matrix where each vectorized image is a column. An eigenvalue decomposition via Principal Component Analysis (PCA) is performed on the covariance of this matrix. The background is then represented by the most descriptive eigenvectors that encompass all possible illuminations to decrease sensitivity to illumination. Several improvements of the PCA approach have been proposed. To name a few, Xu *et al.* [188] proposed a variation of the eigen background model which includes a recursive error compensation step for more accurate detection. Others [150, 96] proposed PCA methods with computationally efficient background updating scheme, while Doug *et al.* [36] proposed illumination invariant approach based on a multi-subspace PCA, each subspace representing different lighting conditions.

Instead of the conventional background and foreground definition, Porikli [136] decomposes a image into "intrinsic" background and foreground images. The multiplication of these images reconstructs the given image. Inspired by the sparseness of the intensity gradient, it applies a spatial derivative filters in the log domain to a subset of the previous video frames to obtain intensity gradient. Since the foreground gradients of natural images are Laplacian distributed and independent, the ML estimate of the background gradient can be obtained by a median operator and the corresponding foreground gradient is computed. These gradient results are used to reconstruct the background and foreground intensity images using a reconstruction filter and inverse log operator. This intrinsic decomposition is shown to be robust against sudden and severe illumination changes, but it is computationally expensive.

Another background subtraction approach based on the theory of sparse representation and dictionary learning is proposed in [194]. This method makes the following two important assumptions: (1) the background of a scene has a sparse linear representation over a learned dictionary; (2) the foreground is sparse in the sense that a majority of the pixels of the frame belong to the background. These two assumptions enable handling both sudden and gradual background changes.

1.2.6 Other Methods

Prediction Models

Early approaches use filters to predict background pixel intensities (or colors). For these models, each pixel whose observed color is far from its prediction is assumed to indicate motion. In [78] and [198], a Kalman filter is used to model background dynamics. Similarly, in [168] Wiener filtering is used to make a linear prediction at pixel level. The main advantage of these methods are their ability to cope with background changes (whether it is periodic or not) without having to assumed any parametric distribution.

In [168] camouflage artifacts and small motionless regions (usually associated to homogeneous foreground regions) are filled in within a post-processing stage. In case most of the pixels in a frame exhibit sudden change, the background models are assumed to be no longer valid at frame level. At this point, either a previously stored pixel-based background model is swapped in, or the model is reinitialized.

Motion Segmentation

Motion segmentation refers to the assignment of groups of pixels to various classes based on the speed and direction of their movements [102]. Most approaches to motion segmentation first seek to compute optical flow from an image sequence. Discontinuities in the optical flow can help in segmenting images into regions that correspond to different objects. In [180], temporal consistency of optical flow over a narrow time window is estimated; areas with temporally-consistent optical flow are deemed to represent moving objects and those exhibiting temporal randomness are assigned to the background.

Optical flow-based motion detection methods will be erroneous if brightness constancy or velocity smoothness assumptions are violated. In real imagery, such violations are quite common. Typically, optical flow methods fail in low-texture areas, around moving object boundaries, at depth discontinuities, etc. Due to the commonly imposed regularization term, most optical flow methods produce an over smooth optical flow near boundaries. Although solutions involving a discontinuity preserving optical flow function and object-based segmentation have been proposed [114], motion segmentation methods usually produce a

P.-M. JODOIN, S. PIÉRARD, Y. WANG, and M. VAN DROOGENBROECK. **Background Modeling and Foreground Detection for Video Surveillance**. In *T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, editors, "Background Modeling and Foreground Detection for Video Surveillance",* chapter 24, CRC Press, July 2014.

halo artifact around moving objects. The resulting errors may propagate across the entire optical flow solution. As a solution, some authors [126, 200] use motion segmentation and optical flow in combination with a color-based GMM model.

Machine Learning

Motion detection methods in this category use machine learning discriminative tools such as SVM and neural networks to decide whether or not a pixel is in motion. The parameters of these functions are learned given a training video. Lin *et al.* [100] uses a probabilistic SVM to initialize the background model. They use the magnitude of optical flow and inter-frame image difference as features for classification. Han and Davis [60] model the background with kernel density approximation with multiple features (RGB, gradient, and Haar) and use a Kernel-SVM as a discriminative function. A somewhat similar approach has also been proposed by Hao [62]. These approaches are typical machine learning methods that need positive and negative examples for training. This is a major limitation for any practical implementation since very few videos come with manually labeled data. As a solution, Chen *et al.* [32] proposed a GPU-based 1-class SVM method called SILK. This method do not need pre-labeled training data, but also allows for online updating of the SVM parameters. Maddalena and Petrosino [104, 106] model the background of a video with the weights of a neural network. A very similar approach but with a post-processing MRF stage has been proposed by Schick *et al.* [152]. Results reported in the paper show great compromise between processing speed and robustness to noise and background motion.

1.2.7 Features

Several features can be used to detect moving objects. The simplest one is certainly grayscale (or luminance) which is easy to interpret and has a well founded physical meaning [50]. Grayscale motion detection methods are normally used on mono-channel cameras like depth cameras, thermal cameras, or older grayscale surveillance cameras.

Nowadays, most motion detection methods rely on color. A color image consists of three channels per pixel (typically R, G, B) that can be processed separately or simultaneously. However, the physical meaning of these channels is less obvious than for mono-channel sensors. Ideally, color images are acquired using three spatially aligned sensors. But since this configuration increases the size and cost of the sensor and requires pixel registration, most color cameras use a single image sensor with a color filter array in front of it. The most widely implemented array is the Bayer color filter array [10]. Each location on the sensor measures one color and missing colors are interpolated from neighboring pixels. Suhr [160] proposes a GMM variant that conducts background modeling in a Bayer-pattern domain and foreground classification in an interpolated RGB domain. The authors argue that since performance is similar to that of the original GMM on RGB images, RGB video streams captured with one sensor are not 3 times more informative than their grayscale counterpart.

In practice though, most techniques exhibit a small performance increase for the classification task when using RGB instead of grayscale features [12]. Thus, from a classification perspective and despite that the computation time is more or less tripled, it is beneficial to use color images, even when colors have been interpolated in the image. In their survey paper, Benezeth *et al.* [12] compare six RGB color distance functions used for background subtraction, including the Euclidean distance, the L1 distance, and the Mahalanobis distance. They conclude that four of the six metrics had globally similar classification performances; only the simplest zero and first order distances were less precise.

Several motion detection techniques use other color spaces such as normalized color [116], cylindric color model [82], HSV [34], HSI [176], YCbCr [88], and normalized RGB [186]. From an application perspective, those colorspaces are believed to be more robust to shadows and illuminations changes than RGB or grayscale [34].

Other features, like edges [68], texture [98], optical flow [100, 116, 164], PCA-based features [124] are also used. Like the colorspace features, these features seem more robust to illumination changes and shadows than RGB features. Texture and optical flow features are also robust to noise and background motion. Since texture features integrates spatial information which often happens to be constant, a slight variation of in the background does not lead to spurious false positives. For example, a bush with a uniform texture will be undetected when shaken by the wind. As for optical flow, since moving objects are assumed to have a smooth and coherent motion distribution [164], noise and random background motion can be easily decorrelated from actual moving objects.

In general, it seems like adding features improves performances. Parag *et al.* [128] even propose to select the best combination of features at each pixel. They argue that different parts of the image may have different statistics and thus require different features. But this comes at the price of both a complexity and a computation time increase.

1.2.8 Updating Strategies

In order to produce consistent results over time, background models need to be updated as the video streams in. From a model point of view, there are two major updating techniques [132] : the *recursive* and *non-recursive* techniques. The *recursive* techniques maintain a single background model that is updated with each new video frame. *Non-recursive* techniques maintain a buffer L of n previous video frames and estimate a background model based solely on the statistical properties of these frames. This includes median filtering and eigenbackgrounds [124]. The major limitation of this last approach is that computing the basis functions requires video clips void of foreground objects. As such, it is not clear how the basis functions can be updated over time if foreground objects are continuously present in the scene.

As mentioned by Elgammal *et al.* [44], other updating strategies use the output of the segmentation process. The *conditional* approach (also called *selective* or *conservative*) updates only background pixels in order to prevent the background model from being corrupted by foreground pixels. However, this approach is incapable of eliminating false positives as the background model will never adapt to it. Wang *et al.* [174] propose to operate at the blob level and define a mechanism to incorporate pixels in the background after a given period of time. As an alternative, the *unconditional* (or *blind*) approach updates every pixel whether it is identified as being active or not. This approach has the advantage of integrating new objects in the background and compensating for false detections caused, say, by global illumination changes or camera jitter. On the other hand, it can allow slowly moving objects to corrupt the background which leads to spurious false detections. Both conditional and unconditional techniques can be used, depending on the appropriateness to the model or on the requirements of the application.

Some authors introduce more nuances. For example, Porikli *et al.* [138] define a GMM method and a Bayesian updating mechanism, to achieve accurate adaptation of the models. A somewhat similar refinement method is proposed by Van Droogenbroeck *et al.* [170]. Both [138] and [170] distinguish between a *segmentation* mask, the binary output image which corresponds to the background/foreground classification result, and the *updating* mask. The updating mask corresponds to locations indicating which pixels have to be updated. The updating mask differs from the segmentation map in that it remains unknown to the user, and depends on updating strategies. For example, one can decide not to update the model inside of static blobs or, on the contrary, decide to erode foreground mask to progressively remove ghosts. Another recent updating strategy consists in spatial diffusion; it was introduced with ViBe [6]. Spatial diffusion is a mechanism wherein a background value is diffused in a neighboring model. This diffusion mechanism can be modulated to

P.-M. Jodoin, S. Piérard, Y. Wang, and M. Van Droogenbroeck. **Background Modeling and Foreground Detection for Video Surveillance**. In *T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, editors, "Background Modeling and Foreground Detection for Video Surveillance",* chapter 24, CRC Press, July 2014.

help remove ghosts or static objects.

1.2.9 Spatial Aggregation, Markovian Models and Post-processing

Most motion detection techniques are local processes that focus on pixel-wise statistics and ignoring neighboring pixels (at least during the modeling phase). This is a well-founded approach from a statistical point of view since neighboring pixels might have very different underlying feature probability density functions. Nevertheless, there exist techniques that aggregate information from neighboring pixels into regular blocks or so-called superpixels. Block-based aggregation is a coherent approach for video encoder, as blocks and macro-blocks are the fundamental spatial units in encoders.

Grouping pixels in blocks is motivated by several factors. First, statistics averaged over a rectangular region increases the robustness to non-stationary backgrounds, despite the fact that it blurs the object silhouette and that another method might be needed to refine edges as in [30]. Second, if sharp edges are not mandatory, processing blocks speeds up the motion detection process. Hierarchical methods, as proposed by Park et al. [130] or Chen et al. [28], are typical examples of methods that plays with different levels of pixel aggregation.

Pixels aggregation can also be achieved by with the help of a Markovian model. Typical Markovian models are based on a maximum a posteriori formulation that is solved through an optimization algorithm such as iterative optimization scheme (ICM) or graphcut [2, 112] which are typically slow. In [14] it was shown that simple Markovian methods (typically those using the Ising prior) produce similar results than simple post-processing filters.

Other Markovian methods have been proposed. In [66], Markov random fields are used to re-label pixels. First, a region-based motion segmentation algorithm is developed to obtain a set coherent regions. This serves to define the statistics of several Markovian random fields. The final labeling is obtained by maximizing the a posteriori energy of the Markov random fields, which can be seen as a post-processing step. The approach by Schick et al. [152] relies on similar ideas. A first segmentation is used to define a probabilistic superpixel representation. Then a post-processing is applied on the statistical framework to provide an enhanced segmentation map. It is interesting to note that Schick et al. [152] have successfully applied their post-processing technique to several motion detection techniques.

A more classical, simpler and faster way to re-label pixels is throughout a post-processing filter. For example, Parks and Fels [132] consider a number of post-processing techniques to improve the segmentation map. Their results indicate that the performance is improved by morphological filters (closings), noise removal filter (such as median filters), and area filters. Morphological filters are used to fill internal holes and small gaps, while area filters are used to remove small objects.

In section 1.4.2, we present the results of some post-processing operations. It appears that simple post-processing operations, such as the median or close/open morphological operations always improve the segmentation map. It is thus recommended to include post-processing operations, even when comparing techniques. This was also the conclusion of Brutzer et al. [24] and Benezeth et al. [14]. Note that other filters can be used such as temporal filters, shadow filters [24], and complex spatio-temporal filtering techniques to re-label the classification results. However, it has been observed that not all post-processing filters do improve results [132].

1.3 Datasets and Survey Papers

Without aiming to be exhaustive, we list below 15 of the most widely used datasets for motion detection validation (see Table 1.2). Additional details regarding some of these

P.-M. Jodoin, S. Piérard, Y. Wang, and M. Van Droogenbroeck. **Background Modeling and Foreground Detection for Video Surveillance**. In *T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, editors, "Background Modeling and Foreground Detection for Video Surveillance",* chapter 24, CRC Press, July 2014.

Dataset	**Description**	**Ground truth**
2012 CD.net*	31 videos in 6 categories : baseline, dynamic background, camera jitter, shadow, intermittent motion, and thermal.	Pixel-based labeling of 71,000 frames.
Wallflower [168]	7 short video clips, each representing a specific challenge such as illumination change, background motion, etc	Pixel-based labeling of one frame per video.
PETS [192]	Many videos aimed at evaluating the performance of tracking algorithms	Bounding boxes.
CAVIAR*	80 staged indoor videos representing different human behaviors such as walking, browsing, shopping, fighting, etc.	Bounding boxes.
i-LIDS*	Very long videos meant for action recognition showing parked vehicle, abandoned object, people walking in a restricted area, and doorway	Not fully labeled.
ETISEO*	More than 80 videos meant to evaluate tracking and event detection methods.	High-level label such as bounding boxes, object class, event type, etc.
ViSOR 2009* [172]	Web archive with more than 500 short videos (usually less than 10 seconds)	Bounding boxes.
BEHAVE 2007*	7 videos shot by the same camera showing human interactions such as walking in group, meeting, splitting, etc.	Bounding boxes.
VSSN 2006*	9 semi-synthetic videos composed of a real background and artificially-moving objects. The videos contain animated background, illumination changes and shadows, however they do not contain any frames void of activity.	Pixel-based labeling of each frame.
IBM *	15 videos taken from PETS 2001 plus additional videos.	Bounding box around each moving object in 1 frame out of 30.
Karlsruhe*	4 grayscale videos showing traffic scenes under various weather conditions.	10 frames per video have pixel-based labeling.
Li *et al.*[94]*	10 small videos (usually 160×120) containing illumination changes and dynamic backgrounds.	10 frames per video have pixel-based labeling.
Karaman *et al.* [76]	5 videos coming from different sources (the web, the "art live" project*, etc.) with various illumination conditions and compression artifacts	Pixel-based labeling of every frame.
cVSG 2008 [166]*	15 Semi-synthetic videos with various levels of textural complexity, background motion, moving object speed, size and interaction.	Pixel-based labeling obtained by filming moving objects (mostly humans) in front of a blue-screen and then pasted on top of background videos.
Brutzer *et al.* [24]	Computer-generated videos showing one 3D scene representing a street corner. The sequences include illumination changes, dynamic background, shadows, and noise.	Pixel-based labeling.

TABLE 1.2 Overview of 15 video datasets.

datasets can be found on a web page of the European CANTATA project*.

Out of these 15 datasets, 7 were initially made to validate tracking and pattern recognition methods (namely PETS, CAVIAR, i-LIDS, ETISEO, ViSOR 2009, BEHAVE 2007, IBM). Although challenging for these applications, those datasets mostly contain day-time videos with fixed background, constant illumination, few shadows and no camera jitter. As a consequence, it is difficult to evaluate how robust motion detection methods are when looking at benchmarking results reported on these 7 datasets.

CD.net* [54] is arguably the most complete dataset devoted to motion detection benchmarking. It contains 31 videos in 6 categories with nearly 90,000 frames, most of it manually labeled. A unique aspect with this dataset is its web site which allows for performance evaluation and method ranking. As of today, 27 methods are reported on the website. CD.net replaced older datasets which had either few videos, partly labeled videos, very short video clips, or semi-synthetic (and yet not realistic) videos.

In parallel of these datasets, a number of survey papers have been written on the topic of motion detection. In this chapter, we list survey papers devoted to the comparison

*www.hitech-projects.com/euprojects/cantata/datasets_cantata/

*www.changedetection.net

P.-M. Jodoin, S. Piérard, Y. Wang, and M. Van Droogenbroeck. **Background Modeling and Foreground Detection for Video Surveillance**. In *T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, editors, "Background Modeling and Foreground Detection for Video Surveillance",* chapter 24, CRC Press, July 2014.

Survey	Description and Benchmark
Goyette *et al.*, 2012 [54]	Survey paper written in the wake of the CVPR 2012 Change Detection workshop. It surveys several methods and reports benchmark results obtained on the CD.net dataset.
Bouwmans *et al.*, 2011 [18]	Probably the most complete surveys to date with more than 350 references. The paper reviewed methods spanning 6 motion detection categories and the features used by each method. The survey also listed a number of typical challenges and gave insights into memory requirements and computational complexity. Benchmark on the Wallflower dataset.
Brutzer *et al.*, 2011 [24]	Report benchmarking results for 8 motion detection method on the computer-generated Brutzer dataset.
Benezeth *et al.*, 2010 [14]	Used a collection of 29 videos (15 camera-captured, 10 semi-synthetic, and 4 synthetic) taken from PETS 2001, the IBM dataset, and the VSSN 2006 dataset.
Bouwmans *et al.*, 2008 [20]	Survey of GMM methods. Benchmarking has been done on the Wallflower dataset.
Parks and Fels, 2008 [132]	Benchmark results for 7 motion detection methods and evaluation of the influence of post-processing on their performance. They used 7 outdoor and 6 indoor videos containing different challenges such as dynamic backgrounds, shadows and various lighting conditions.
Bashir and Porikli, 2006 [8]	Performance evaluation of tracking algorithms using the PETS 2001 dataset by comparing the detected bounding box locations with the ground-truth.
Nascimento and Marques, 2006 [120]	Report benchmarks obtained on a single PETS 2001 video with pixel-based labeling.
Radke *et al.* [144]	Extensive survey of several motion detection methods. Most of the discussion in the paper was related to background subtraction methods, pre- and post-processing, and methodologies to evaluate performances. Contains no quantitative evaluation.
Piccardi [134]	Reviewed 7 background subtraction methods and highlighted their strengths and weaknesses. Contains no quantitative evaluation.
Rosin and Ioannidis, 2003 [148]	Report results for 8 methods. Videos used for validation show two lab scenes with balls rolling on the floor.
Prati *et al.*, 2001 [142]	Used indoor sequences containing one moving person. 112 frames were labeled.

TABLE 1.3 15 motion detection surveys.

and ranking of motion detection algorithms. Note that some of these surveys use datasets mentioned previously while other use their own dataset. Details can be found in Table 1.3.

These survey papers often contain a good overview of state-of-the-art motion detection method. However, the reader shall keep in mind that statistics reported in some of these papers were not computed on a well-balanced dataset composed of real (camera-captured) videos. Typically, synthetic videos, real videos with synthetic moving objects pasted in, or real videos out of which only 1 frame was manually segmented for ground truth were used. Also, some survey papers report results from fairly simple and old motion detection methods.

1.4 Experimental Results

So far, we introduced eight families of motion detection methods, presented different features, several updating schemes and many spatial aggregation and post-processing methods. The goal of this section is to provide empirical results to validate which configuration performs best. Note that since the number of combinations of motion detection methods, features, updating schemes and post-processing methods is intractable, we provide benchmarks for each aspect independently.

The goal of this section is also to underline unsolved issues in motion detection and identify complementary methods whose combination can further improve results. Empirical results are obtained with the 2012 CD.net dataset. As mentioned previously, this dataset includes 31 videos divided in 6 categories namely dynamic background, camera jitter, shadow, intermittent motion, baseline, and thermal. With a manually labeled groundtruth for each video, to our knowledge this is the most complete dataset for motion detection validation.

But prior to present benchmarking results, we first describe and explain the evaluation metrics used in this section.

1.4.1 Metric Evaluation

As stated by Goyette *et al.* [56], it is not a trivial task to find the right metric to accurately measure the ability of a method to detect motion. If we consider background segmentation as a classification process, then we can recover the following 4 quantities for a processed video: the number of true positives (TP) and false positives (FP), which accounts for the number of foreground pixels correctly and incorrectly classified, and the number of true negatives (TN) and false negatives (FN), which are similar measures but for background pixels. With these values, one can come out with the following seven metrics, often used to rank background subtraction methods. (1) The *True Positive Rate* (TPR), also named *sensitivity* and *recall*, is $Re = TPR = \frac{TP}{TP+FN}$ (2) The *False Negative Rate (FNR)* is $FNR = 1 - TPR$. (3) The *True Negative Rate* (TNR), also named *specificity*, is $TNR = \frac{TN}{TN+FP}$. (4) The *False Positive Rate* is $FPR = 1-TNR$. (5) The *precision* is $Pr = \frac{TP}{TP+FP}$. (6) The *Probability of Wrong Classification* (also Error Rate) is $PWC = \frac{FP+FN}{TP+TN+FP+FN}$. (7) The *Accuracy* is $A = 1 - PWC$.

In the upcoming subsections, we will try to answer the question of which metric(s) should be used to rank methods.

Limitation of metrics combining TPR and TNR.

For obvious reasons, TPR, TNR, FPR, and FNR cannot be used alone to rank methods. In fact, methods are typically adjusted to prevent FPR and FNR from being large simultaneously. Such trade-offs can be interpreted by showing a Receiver Operating Characteristic (ROC) graph. But ranking methods based on ROC curves is rather inconvenient due to the large number of results that need to be generated and which can be prohibitive in the context of large videos. Therefore, most often, only a single point is known in the ROC space. Summarizing TPR and TNR into a single value remains difficult and this is highlighted by the following discussion.

Since most surveillance videos exhibit a low amount of activity (5 % on average for the CD.net video sequences), the TNR value will always dominate A and PWC. Actually, as one can see in Table 1.4, nearly all methods have a very low FPR (except for [70]) and a large FNR. As a consequence, when used alone, the accuracy A and the probability of wrong classification PWC will always favor methods with a low FPR and a large FNR. At the limit, a method that would detect no moving object at all would have a not-so-bad ranking score according to A and PWC alone. That is because only a small fraction of the pixels would be wrongly classified on average.

Another way of underlying the limitation of A (and PWC) is by rewriting the accuracy equation. If we denote the probabilities of observing a foreground pixel and a background pixel by p_{FG} and p_{BG} respectively, then one can show that the accuracy can be computed as follows $A = p_{FG} TPR + p_{BG} TNR$. Thus, with a low p_{FG}, the TNR ends up having an overwhelming importance when computing A. As an alternative, one could consider the *Balanced Accuracy*, $BA = \frac{1}{2} TPR + \frac{1}{2} TNR$. However, since that metric is uncommon in the motion detection community, we decided not to use it.

So in conclusion, although the accuracy and the probability of wrong classification can be used to evaluate methods, they should not be used alone and one should keep in mind that they favor methods with a low FPR.

Metrics Derived from Pr and Re

Another trade-off for motion detection methods is to prevent Pr and Re from being large simultaneously, which is shown on a precision-recall curve.

P.-M. Jodoin, S. Piérard, Y. Wang, and M. Van Droogenbroeck. **Background Modeling and Foreground Detection for Video Surveillance**. In *T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, editors, “Background Modeling and Foreground Detection for Video Surveillance”,* chapter 24, CRC Press, July 2014.

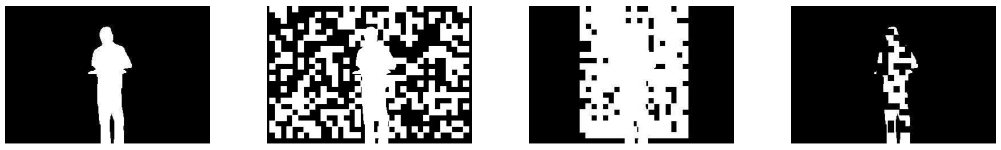


FIGURE 1.1 Three methods with the same balanced accuracy (0.8) but with different F-Measures. For methods 1 and 2, $F_1 = 0.35$ while for method 3 $F_1 = 0.73$.

The classical difficulties encountered when interpreting those graphs are mainly related to an unachievable region in the precision-recall space [22]. There exists a lower bound on the precision which depends on the recall and p_{FG}. The size of this region grows with p_{FG}, and the precision varies significantly with p_{FG} [92]. Moreover, a purely random method is such that $Pr = p_{FG}$, and therefore the range for valid Pr is not the same for all videos. Therefore, the metrics derived from Pr and Re should be interpreted with care.

But using precision-recall curves to rank methods is inconvenient for the same reasons ROC curves are. In practice, precision and recall must be combined into one metric. The most frequent way of doing so is through the F-measure F_1, which is the harmonic mean between Pr and Re: $F_1 = \left(\frac{1}{2}Pr^{-1} + \frac{1}{2}Re^{-1}\right)^{-1} = \frac{2\,TP}{2\,TP+FP+FN}$. When both Pr and Re are large, F_1 is approximately equal to the arithmetic mean of Pr and Re. Otherwise, it is approximately equal to $\min\left(Pr,\, Re\right)$.

The balanced accuracy and the F-Measure, although similar at the first glance, are not equivalent. Let's consider the case shown in Figure 1.1 for which it is difficult to identify the human silhouette based on the first two results. In that example, all three results have the same balanced accuracy but a much higher F-Measure for method 3. This is a strong indication that the F-measure is a better metric than the balanced accuracy in the context of motion detection and thus why we use it in our validation. Another reason for F_1 to be larger for method 3 is the fact that it does not take into account TN. As a consequence, F_1 is a metric that focuses more on the foreground than on the background.

Influence of Noise

The F-measure is not void of limitations. As will be shown in this section, it is sensitive to noise and thus should be used with care. In order to illustrate the impact of noise and the importance of post-filtering operations, let us add a “salt and pepper” noise to a segmentation map. Let α be the probability to switch the class of a pixel and TPR' and TNR' the estimates on the noisy segmentation maps. In that case, we have $TP' = TP\left(1-\alpha\right) + FN\alpha$, $FN' = FN\left(1-\alpha\right) + TP\alpha$, $TN' = TN\left(1-\alpha\right) + FP\alpha$, and $FP' = FP\left(1-\alpha\right) + TN\alpha$. Following some algebric manipulations, one can show that the relative ranking between 2 methods can change depending on the amount of noise in the data. This is illustrated in Figure 1.2 where F_1 for Spectral-360 goes below PBAS after noise has been added.

This sensitivity to noise leads us to conclude that it is preferable to filter noise with a post-processing filter before ranking background subtraction techniques according to F_1. This is what has been done for every method reported in Table 1.4.

Interestingly, the ranking provided by PWC, the accuracy and the balanced accuracy is not affected by noise as is the case for the F-Measure. Here is why. After some algebraic manipulations, we find that $TPR' = \frac{TP'}{TP'+FN'} = \left(1-\alpha\right)TPR + \alpha\left(1-TPR\right)$, and $TNR' = \frac{TN'}{TN'+FP'} = \left(1-\alpha\right)TNR + \alpha\left(1-TNR\right)$. If $\alpha < \frac{1}{2}$, then the same amount of noise on the results of two segmentation algorithms does not change their respective ranking, if the

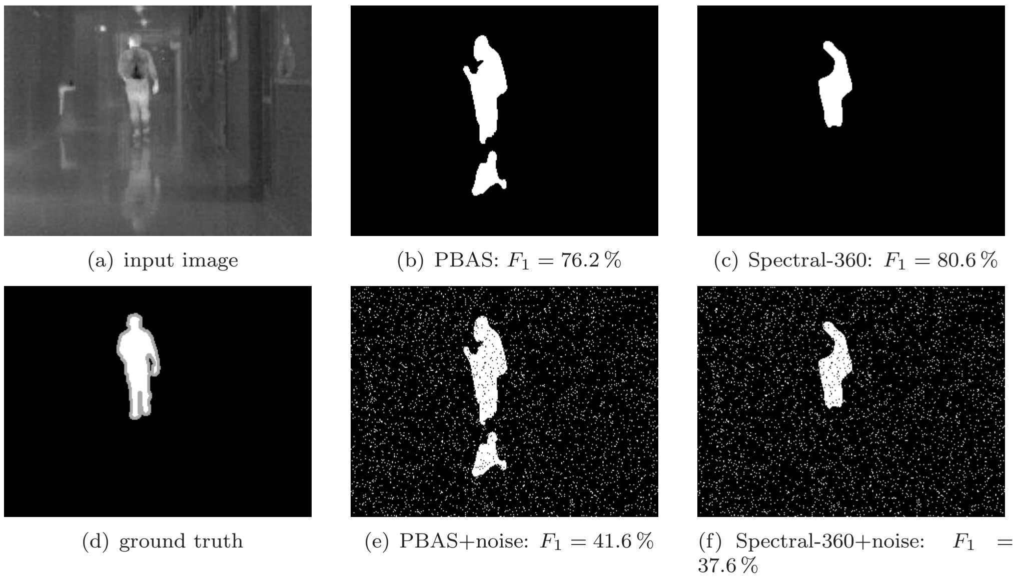

(a) input image (b) PBAS: $F_1 = 76.2\,\%$ (c) Spectral-360: $F_1 = 80.6\,\%$ (d) ground truth (e) PBAS+noise: $F_1 = 41.6\,\%$ (f) Spectral-360+noise: $F_1 = 37.6\,\%$

FIGURE 1.2 Ranking of the methods obtained according to F-measure is sensitive to noise. It is therefore important to filter out noise from the results before ranking methods with F_1.

ranking is determined by any linear combination Q of TPR and TNR. Indeed, $Q = \beta TPR + (1-\beta) TNR$ and $Q' = \beta TPR' + (1-\beta)\, TNR' = (1-\alpha)\, Q + \alpha\,(1-Q)$. Thus $Q_{alogoritm\,1} \leq Q_{algorithm\,2}$ if and only if $Q'_{algorithm\,1} \leq Q'_{algorithm\,2}$. This implies that the ranking according to the accuracy, the balanced accuracy, or the probability of wrong classification is stable as long as the same amount of noise is present on the output of the compared methods.

Evaluation and Ranking of Methods

The previous discussion made it clear that summarizing the performance of a background subtraction algorithm with a single metric is restrictive. Several metrics like FNR and FPR are complementary and cannot be used independently whereas others like PWC and A give an overwhelming importance to TNR. As for the F-measure, although widely used, it is sensitive to noise. This leads us to conclude that no metric is perfect and should thus be used with care.

The last question that we ought to answer before presenting benchmark results, is how to compute evaluation metrics when considering more than one video sequence. Naively, one could add up the total number of TP, TN, FP and FN across all videos out of which metrics could be computed. But unfortunately, since videos have different sizes in space and time, large videos would end up having more influence on the ranking that smaller ones. As explained by Goyette *et al.* [56], a better solution is to compute the metrics for each video (that is $Re, FNR, FPR, Specificity, Pr, PWC, A$, and F_1) and than average it across videos. CD.net also has a multi-criteria ranking which we do not retained in this chapter for the sake of simplicity.

In this chapter, we rank methods according to the average F_1 computed across all videos and categories of the CD.net dataset. Although sensitive to noise, the result of every method has been post-processed with a median filter to prevent the previously-mentioned ranking problems. Also, Goyette *et al.* [56] mentioned that the F_1 score is correlated with this multi-criteria ranking which is a good indication that F_1 is a well balance metric.

P.-M. Jodoin, S. Piérard, Y. Wang, and M. Van Droogenbroeck. **Background Modeling and Foreground Detection for Video Surveillance**. In *T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, editors, “Background Modeling and Foreground Detection for Video Surveillance”,* chapter 24, CRC Press, July 2014.

Method	Description	FPR	FNR	F-Measure
Spectral-360 [154]	Patent	0.008	0.22	0.77
DPGMM [58]	GMM + Dirichlet Process	0.014	0.17	0.77
SGMM-SOD [46]	Improved version of SGMM [48]	0.006	0.23	0.76
PBAS [64]	data-driven and stochastic method	0.010	0.21	0.75
PSP-MRF [152]	Probabilistic super-pixels with Neural Maps	0.017	0.19	0.73
SC-SOBS [106]	Improved version of SOBS [104]	0.016	0.19	0.72
SOBS [104]	Neural maps	0.018	0.21	0.71
SGMM [48]	GMM + new mode initialization, updating and splitting rule	0.009	0.29	0.70
Chebyshev Inequality [118]	Multistage method with Chebyshev inequality and object tracki	0.011	0.28	0.70
KNN [202]	Data-driven KNN	0.009	0.32	0.67
KDE Elgammal [42]	Original KDE	0.024	0.25	0.67
GMM Stauffer-Grimson [156]	Original GMM	0.014	0.28	0.66
GMM Zivkovic [202]	GMM with automatic mode selection	0.015	0.30	0.65
KDE Yoshinaga *et al.* [190]	Spatio-temporal KDE	0.009	0.34	0.64
KDE Nonaka *et al.* [122]	Multi-level KDE	0.006	0.34	0.64
Bayesian Multi layer [138]	Bayesian layers + EM	0.017	0.39	0.62
Mahalanobis distance [14]	Basic background subtraction	0.040	0.23	0.62
Euclidean distance [14]	Basic background subtraction	0.030	0.29	0.61
GMM KaewTraKulPong [74]	Self-adapting GMM	0.005	0.49	0.59
Histogram over time [196]	Basic method with color histograms	0.065	0.23	0.54
GMM RECTGAUSS-Tex [146]	Multiresolution GMM	0.013	0.48	0.52
Local-Self similarity [70]	Basic method with self- similarity measure	0.148	0.06	0.50

TABLE 1.4 Overall results for 22 methods. These results correspond to the average FPR, FNR and $F-Measure$ obtained on all 31 videos of the CD.net dataset.

Let us mentioned that the benchmarking results do not entirely capture the pros and cons of a method. Obviously, the complexity of an algorithm together with its processing speed and memory usage are to be considered for real-time applications.

1.4.2 Benchmarks

Motion Detection Methods

From the changedetection.net website, we retained results from 22 motion detection methods. Five methods are relatively simple as they rely on plain background subtraction, of which two use color features (Euclidean and Mahalanobis distance methods described in [18, 14]), one uses RGB histograms over time [196], and one uses local self-similarity features [70].

We also report results for eight parametric methods, seven of which use a GMM model. This includes the well-known methods by Stauffer and Grimson [156], a self-adapting GMM by KaewTraKulPong [74], the improved GMM method by Zivkovic and Heijden [202], the multiresolution block-based GMM (RECTGAUSS-Tex) by Dora *et al.* [146], GMM method with a Dirichlet process (DPGMM) that automatically estimated the number of Gaussian modes [58] and the SGMM and SGMM-SOD methods by Evangelio *et al.* [48, 46] which rely on a new initialization procedure and novel mode splitting rule. We also included a recursive per-pixel Bayesian approach by Porikli and Tuzel [138] which shows good robustness to shadows according to [54].

We also report results on three KDE methods. The original method by Elgammal *et al.* [42], a multi-level KDE by Nonaka *et al.* [122], and a spatio-temporal KDE by Yoshinaga *et al.* [190]. Results for data-driven methods and machine learning methods are also reported. That is Hofmann's stochastic and self-adaptive method (PBAS) [64], a simple K-nearest neighbor method [202] and neural maps methods (SOBS and SC-SOBS) by Maddalena *et al.* [104, 106] and a neural network method with a region-based Markovian post-processing methods (PSP-MRF) by Schick *et al* [152]. We also have results for two commercial products. One that does pixel-level detection using the Chebyshev inequality and peripheral and recurrent motion detectors by Morde *et. al.* [118] and one which has only been published in a pending patent so far and whose description is not available [154]. The false positives rate (FPR), false negative rate (FNR) and F-measure (F_1) for these 22 methods are reported in Table 1.4. Note that, as mentioned in [54], these are the *average* FPR, FNR and F_1 across all videos.

Category	1st		2nd		3rd	
Baseline	SC-SOBS	0.93	Spectral-360	0.93	PSP-MRF	0.92
Dynamic Back.	DPGMM	0.81	Spectral-360	0.79	Chebyshev Inequality	0.77
Shadows	Spectral-360	0.88	SGMM-SOD	0.86	PBAS	0.86
Camera Jitter	PSP-MRF	0.78	DPGMM	0.75	SGMM	0.75
Thermal	DPGMM	0.81	Spectral-360	0.78	PBAS	0.76
Interm. Motion	SGMM-SOD	0.72	SC-SOBS	0.59	PBAS	0.58

TABLE 1.5 Three highest ranked method for each category together with their F-measure.

From these results, one can conclude that the top performing methods are mostly GMM methods (DPGMM, SGMM-SOD, SGMM), data-driven methods (KNN and PBAS) and machine learning methods (SOBS and PSP-MRF). As shown in Table 1.5, GMM methods (particularly DPGMM and SGM-SOD) seems robust to background motion, camera jitter and intermittent motion. This can be explained by the fact that these GMM methods come with a mode initialization (and updating) procedures that reacts swiftly to changes in the background. Table 1.5 also shows that there is room for improvement on jittery sequences and intermittent motion which are the categories with the lowest F-measure. Another unsolved issue is robustness to shadows. Although the F-measure of the most effective methods is above 0.86, the FPR on shadows of the 3 best methods is above 58%. This means that even the most accurate methods wrongly classify hard shadows.

Features

Here, we report results for 8 of the most commonly-used features *i.e.*: grayscale, RGB, Normalized RGB, HSL, HSV, norm of the gradient, RGB+gradient and YCbCr. We tested these features with two different methods. The first one is a basic background subtraction method with a forgetting constant of 0.002 [14]. The second is a version of ViBe [6] (a stochastic data-driven method) that we adapted to the various color spaces and removed its postprocessing stage.

Results in Table 1.6 leads us to two main conclusions. First, using all three RGB color channels when possible instead of grayscale only always improves results. Second, out of the "illumination-robust" features N-RGB, HSL, HSV and gradient (grad), only HSV seems to provide good results globally. That being said, combining gradient with RGB helps improving results, especially for the basic method. As mentioned by several authors,

Category	**Gray**	**RGB**	**N-RGB**	**HSL**	**HSV**	**grad**	**RGB+grad**	**YCbCr**
Basic Method	0.48	0.53	0.49	0.56	0.58	0.3	0.59	0.59
ViBe [6]	0.72	0.75	0.60	0.65	0.74	0.11	0.74	0.71

TABLE 1.6 F-Measure obtained for 8 different features and two motion detection methods.

this suggests that for some methods, combining color and texture is a good way of improving results.

Updating Scheme

In this section, we tested different updating schemes on one method. We tested the blind, conservative, "soft" conservative and "edge" conservative updating schemes. Again, we implemented a simple background subtraction method with RGB color feature. The difference from one implementation to another is the forgetting constant α. For the blind scheme, $\alpha = 0.002$, for the conservative $\alpha = 0.002$ only for background pixels, the soft conservative $\alpha = 0.002$ for foreground pixels and $\alpha = 0.008$ for background pixels and edge conservative, $\alpha = 0.007$ for background pixel, $\alpha = 0.002$ for foreground edge pixels and $\alpha = 0$ for the other foreground pixels.

Blind	Conservative	Soft-Conservative	Edge-Conservative
0.52	0.5	0.53	0.55

TABLE 1.7 F-measure for different background updating strategies.

Results in Table 1.7 show that the edge-conservative strategy is the most effective one while the conservative strategy is the least effective, although by a small margin. The reason for this small difference between results comes from the fact that the CD.net videos are all relatively short (at most 6 minutes) and thus do not exhibit major changes in the background as is the case when dealing with longer videos. Longer videos would certainly stretch the difference between each strategy.

Post-Processing

In this section, we compared different post-processing filters on the output of 3 methods. These methods are a basic background subtraction method with a forgetting constant of 0.002, ViBe [6] and ViBe+ [170]. Note that ViBe+ is a method which already has a post-processing stage. The post-processing methods are 3 median filters (3×3, 5×5, and 7×7), a morphological opening and closing operation, a closing operation followed by a region filling procedure (as suggested by Parks and Fels [132]) and a connected component analysis. The latter removes small isolated regions, whether they are active regions or not.

Results in Table 1.8 show that all post-processing filters improved the results of all 3 methods. Surprisingly, it also improved the results of ViBe+, a method which already had a post-processing stage! Of course, the improvement rate is more significant for a low ranked method than for a precise one. Given its positive impact on performance and noise removal, we recommend to use at least a 5x5 median, but also other filtering operations to fill gaps, smooth object shapes or remove small regions.

Method	No Post-processing	Med 3×3	Med 5×5	Med 7×7	Morph	Close +fill.	Connected component
Basic Method	0.53	0.56	0.63	0.60	0.55	0.54	0.58
ViBe [6]	0.67	0.68	0.68	0.69	0.70	0.70	0.68
ViBe+ [170]	0.71	0.72	0.73	0.73	0.74	0.74	0.72

TABLE 1.8 F-Measure obtained for 6 different postprocessing filters on the output of 3 motion detection methods.

1.4.3 Combining Methods

So far, we analyzed and compared the behavior of individual motion detection techniques. A further step consists in combining methods. From that point, at least two questions arise: how should methods be combined, and which methods, similar or dissimilar, should be combined?

There are two strategies to combine methods: (1) consider every available methods, regardless of their own performance, or (2) select a small subset of methods, based on their performance or on an optimization criterion. Here, we explore 3 different combination rules : two involving all $n = 22$ methods and one involving a subset of methods. Because it is difficult to model the correlation between individual methods and to take it into account, the combination rules considered here are based on the assumption that individual classifiers are independent of each other. An alternative would be to learn the combination rule [38], but this is out of the scope of this chapter.

The results obtained with the 3 different combination rules are shown on precision recall graphs with F_1 contour lines. It should be noted that the conclusions that can be drawn from the receiver operating characteristic space are different from those of the precision recall space. In this chapter, we only focus on the latter, and aim at maximizing the F_1 score. The following observations should therefore be interpreted with care.

Combination rule 1: majority vote among all methods.

We define a decision thresholding function $\mathcal{F}_{Th}$ as follows: $\mathcal{F}_{Th}(x) = 0$ if $x < Th$, and $\mathcal{F}_{Th}(x) = 1$ otherwise. Let us denote the output of the ith background subtraction method by $\hat{y}_i \in \{0,1\}$, the combined output by $\hat{y}_c \in \{0,1\}$, the ground truth by $y \in \{0,1\}$, and probabilities by $p(\cdot)$. The first combination rule considered in this chapter is

$$\hat{y}_c = \mathcal{F}_{Th}\left(\frac{1}{n}\sum_{i=1}^{n}\hat{y}_i\right), \qquad \text{with } Th \in [0,1]. \tag{1.1}$$

We refer to this technique as the “majority vote” rule, since it extends the classical unweighted majority vote (this one is obtained when n is odd, and the decision threshold is set to $Th = 0.5$). Note that Equation (1.1) defines $n+1$ monotonic functions from $\{0,1\}^n$ into $\{0,1\}$. This combination rule supposes that the individual background subtraction algorithms are independent. Limits of what can be expected from such a combination are discussed in [90]. The results, obtained for every decision threshold, are shown in Figure 1.3(a).

Combination rule 2: summation.

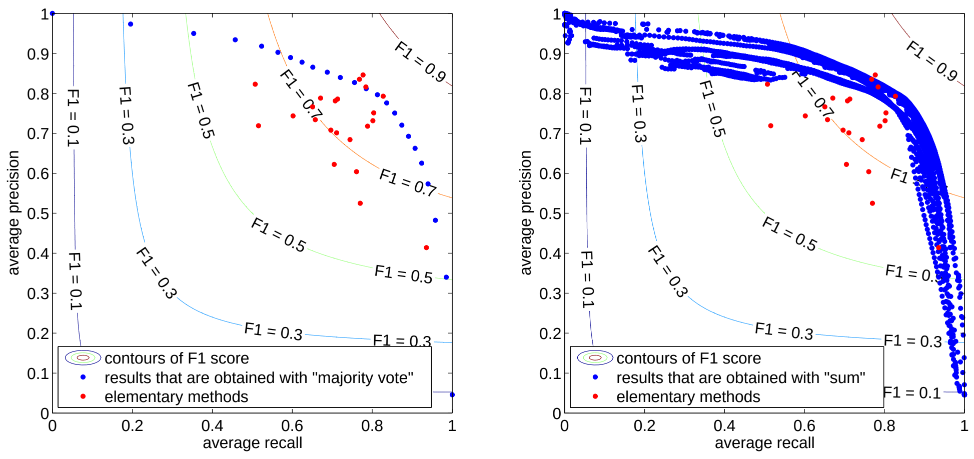
Another combination rule which is often encountered in the literature is the summation rule [86], which is also known as mean rule [162], or the averaged Bayes’ classifier [184]. Adapted to our framework, it can be written as

$$\hat{y}_c = \mathcal{F}_{Th}\left(\frac{1}{n}\sum_{i=1}^{n} p(y=1|\hat{y}_i,\Upsilon)\right), \qquad \text{with } Th \in [0,1], \tag{1.2}$$

where

$$p(y=1|\hat{y}_i=0,\Upsilon) = \frac{FNR_i\, p(y=1|\Upsilon)}{TNR_i\, p(y=0|\Upsilon) + FNR_i\, p(y=1|\Upsilon)}, \tag{1.3}$$

$$p(y=1|\hat{y}_i=1,\Upsilon) = \frac{TPR_i\, p(y=1|\Upsilon)}{FPR_i\, p(y=0|\Upsilon) + TPR_i\, p(y=1|\Upsilon)}. \tag{1.4}$$



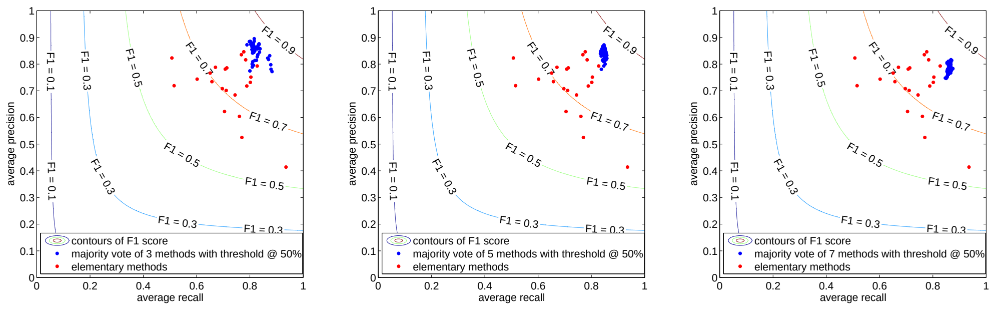
(a) majority vote (b) summation

FIGURE 1.3 Results obtained from the combination of all (22) background subtraction methods, with 2 combination rules. For the purpose of comparison, the precision and the recall of the 22 individual methods are displayed in red. The blue dots correspond to different decision thresholds (Th) as well as different estimations of the priors (Υ).

Here Υ represents the knowledge about the context (Υ is sometimes named the *environment*, as in [184]). The context is, for example, an indoor video-surveillance application, a particular video stream, the other pixels in the same image, or some information related to the past. However, the choice should be carefully made, since it can have an important impact on the performance of the combination. The priors $p\,(y = 0|\Upsilon)$ and $p\,(y = 1|\Upsilon)$ are usually estimated on the basis of the decisions taken by the individual methods on the whole image, in order to adapt dynamically to the context. But, in some video-surveillance settings, some video regions are more likely to contain movement than others. In this case, it makes sense to estimate the priors on a neighborhood around the considered pixel, and also to take the history into account. This is somehow equivalent to the atlas used in [178], but in a dynamic setting.

The results for this combination rule are shown in Figure 1.3(b). We have considered the whole range of decision thresholds, and four ways of estimating the priors: (1) fixed priors ($p\,(y = 1) \in \{4\,\%, 8\,\%, 12\,\%, 16\,\%, 20\,\%\}$); (2) priors estimated on the whole image; (3) priors estimated on the whole image, with a temporal exponential smoothing applied on the estimated priors (with a smoothing factor $\alpha \in \{0.90, 0.37, 0.21, 0.14, 0.09, 0.05, 0.02\}$); (4) priors estimated per pixel, on a square neighborhood of size $s \in \{1, 7, 31, 127, 511\}$. Note that estimating the priors for a combination is an ill-posed problem since false positives (false negatives) tend to increase (reduce) the estimated prior of the foreground, and therefore to encourage a higher number of positives (negatives) in the combined output. Obviously, the opposite behavior is wanted.

We observe some similarities between the majority vote and the summation. However, the majority vote only permits to reach $n = 22$ points in the precision recall space, whereas the summation permits a fine tuning. The optimal threshold for the majority vote and the summation varies significantly from one video to another (this is not represented on the graphs). Thus, there is a trade-off when choosing the threshold. The best overall threshold is about 0.4 for the majority vote and the sum. We have obtained our best results when estimating the priors on a neighborhood of 31×31 pixels.



(a) 50 subsets of 3 methods (b) 50 subsets of 5 methods (c) 50 subsets of 7 methods

FIGURE 1.4 Real precision and recall of the majority vote combination rule (at the neutral decision threshold). The predicted performance is shown, in blue, for 50 combinations of 3, 5, and 7 methods, selected theoretically. The precision and the recall of the 22 individual methods are shown in red.

Combination rule 3: majority vote of a predefined subset of methods.

It turns out that no combination of the 22 methods is able to beat significantly the best individual methods. Carefully selecting a subset of methods is therefore necessary. Note that an alternative would be to assign a "good" weight to each individual background subtraction method.

Our third combination rule is the same as the previous majority vote one, except that it is applied on a subset of 3, 5 and 7 methods. Since computing the majority vote of every possible combination of methods is extremely time consuming, we first determined the 50 most promising subsets of methods. A prediction of the F_1 score has been obtained for every combination of 3, 5 and 7 methods, without the need to try them on the video sequences. This is possible under the assumption of independence, based on the values given in Table 1.4, and the knowledge of the proportion of foreground pixels in each video sequence*.

The results obtained with the third combination rule are depicted in Figure 1.4. We used a decision threshold of 0.5. Whereas a blind combination of all methods together does not permit to beat significantly the best individual methods (see Figure 1.3), combining carefully selected subsets of methods leads to a higher performance than the methods independently (see Figure 1.4).

We have also observed how many times each method appears in the selected subsets of 3, 5, and 7 methods. We have noticed that, as expected, the methods which have the highest F_1 score are often taken into account, even if the ranking in Table 1.4 is not strongly correlated with the occurrences. Surprisingly, about one third of the methods are never selected. What is even more surprising is that the Local-Self similarity method [70], which has the worst ranking according to F_1 in Table 1.4, appears often in the selected combinations for 3

*In a probabilistic framework, the classifier independence means that $p(\hat{y}_1, \ldots, \hat{y}_n | y) = \prod_{i=1}^{n} p(\hat{y}_i | y)$. The proportion of foreground pixels in a video sequence is $p(y=1) = p_{FG}$. Table 1.4 gives $p(\hat{y}_i = 1 | y = 0) = FPR_i$ and $p(\hat{y}_i = 1 | y = 1) = 1 - FNR_i$. Let $\mathcal{C}$ denotes the combination function (the majority vote in this case) such that $\hat{y}_c = \mathcal{C}(\hat{\mathbf{y}})$ with $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_n)$. Under the independence assumption, we expect $p(\hat{y}_c | y) = \sum_{\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_n),\ \text{s.t. } \mathcal{C}(\hat{\mathbf{y}}) = \hat{y}_c} \prod_{i=1}^{n} p(\hat{y}_i | y)$. The predicted precision and accuracy are derived from $p(\hat{y}_c = 1 | y = 1)$ and $p(\hat{y}_c = 0 | y = 0)$ thanks to the knowledge of $p(y=1)$. They are averaged across all videos and categories, before deriving the predicted value of F_1. As we do not expect the independence assumption to hold in practice, the predicted values are only used for the selection of the most promising subsets of methods. The results presented in the figures are obtained experimentally.

methods, and is systematically used in the top 50 subsets of 5 and 7 methods, with no exception. Note that it is not a side effect of the independence assumption, as taking this method into account does not harm to the performance when the errors are positively correlated, as the results shown in Figure 1.4 illustrate. What should be noted about the Local-Self similarity method [70] is that it behaves differently from the other methods: it has the highest TPR, but also the highest FPR. Intuitively, a method that behaves differently may be useful in a combination, even if it has a bad ranking when evaluated alone, thanks to its complementarity with the other methods. This effect has already been observed by Duin *et al.* [40]. Therefore, if combining multiple background subtraction methods is possible, designing methods that are top-ranked, when they are evaluated alone, should not be the primary focus. Instead, designing complementary methods is preferable.

1.5 Conclusion

In this chapter, we presented a survey of 8 families of motion detection methods, presented different features, several updating schemes and many spatial aggregation and post-processing methods. We also provided several benchmarking results based on the CD.net dataset. These results lead us to the following conclusions :

1. **Methods** : As of today, GMM (DPGMM, SGMM-SOD, SGMM), data-driven methods (KNN and PBAS) and machine learning methods (SOBS and PSP-MRF) and among the most effective ones. That being said, none is performing best on every category.
2. **Remaining Challenges** : Intermittent motion, camera jitter and hard shadows are among the most glaring issues.
3. **Features** : HSV and RGB + gradient are the most effective features.
4. **Updating scheme** : The edge-conservative approach is the most effective scheme while the conservative approach is the least effective.
5. **Post-processing** : Every post-processing method that we have tested improved the results of our motion detection methods, especially for the simple low-ranked method. Post-processing should thus always be used.
6. **Combining methods** : One can beat the best performing methods by combining the output of several methods. The best results have been obtained with a majority vote of 3 and 5 methods and with a threshold of 50%. The best results are obtained by not only combining top ranked methods, but by combining methods which are complementary by nature.

References

2. T. Aach and A. Kaup. Bayesian algorithms for adaptive change detection in image sequences using markov random fields. *Signal Process., Image Commun.*, 7:147–160, 1995.
4. M.S. Allili, N. Bouguila, and D. Ziou. Finite general gaussian mixture modelling and application to image and video foreground segmentation. *J. of Elec. Imaging*, 17:1–23, 2008.
6. O. Barnich and M. Van Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. *IEEE Trans. Image Process.*, 20(6):1709–1724, June 2011.
8. F. Bashir and F. Porikli. Performance evaluation of object detection and tracking systems. In *Proc. IEEE Int. Workshop on Performance Evaluation of Tracking Systems*, 2006.
10. B. Bayer. Color imaging array, u.s. patent 3971065, 1976.
12. Y. Benezeth, P. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Review and evaluation of commonly-implemented background subtraction algorithms. In *Proc. Int. Conf. Pattern Recognition*, pages 1–4, December 2008.
14. Y. Benezeth, P-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Comparative study of background subtraction algorithms. *J. of Elec. Imaging*, 19(3):1–12, 2010.
16. A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(3):257–267, 2001.
18. T. Bouwmans. Recent advanced statistical background modeling for foreground detection: A systematic survey. *Recent Patents on Computer Science*, 4(3), 2011.
20. T. Bouwmans, F. El Baf, and B. Vachon. Background modeling using mixture of gaussians for foreground detection: A survey. *Recent Patents on Computer Science*, 1(3):219–237, 2008.
22. K. Boyd, V. Costa, J. Davis, and C. Page. Unachievable region in precision-recall space and its effect on empirical evaluation. In *29th International Conference on Machine Learning (ICML)*, Edinburgh, UK, June-July 2012.
24. S. Brutzer, B. Höferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pages 1937–1944, June 2011.
26. D. Butler, S. Sridharan, and M. Bove. Real-time adaptive background segmentation. In *ICME*, pages 341–344, 2003.
28. S. Chen, J. Zhang, Y. Li, and J. Zhang. A hierarchical model incorporating segmented regions and pixel descriptors for video background subtraction. *IEEE Transactions on Industrial Informatics*, 8(1):118–127, February 2012.
30. Y. Chen, C. Chen, C. Huang, and Y. Hung. Efficient hierarchical method for background subtraction. *Pattern Recognition*, 40(10):2706–2715, 2007.
32. L. Cheng, M. Gong, D. Schuurmans, and T. Caelli. Real-time discriminative background subtraction. *IEEE Trans. Image Process.*, 20(5):1401–1414, 2011.
34. R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(10):1337–1342, October 2003.
36. Y. Dong, T.X. Han, and G.N. Desouza. Illumination invariant foreground detection using multi-subspace learning. *Int. J. Know.-Based Intell. Eng. Syst.*, 14(1):31–41, 2010.
38. R. Duin. The combining classifier: to train or not to train? In *IEEE International Conference on Pattern Recognition (ICPR)*, volume 2, pages 765–770, Quebec

City, Canada, August 2002.

40. R. Duin and D. Tax. Experiments with classifier combining rules. In *Multiple Classifier Systems*, volume 1857 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2000.

42. A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis. Background and foreground modeling using nonparametric kernel density for visual surveillance. *Proc. IEEE*, 90:1151–1163, 2002.

44. A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. *Proc. European Conf. Computer Vision*, pages 751–767, 2000.

46. R. Evangelio, M. Pätzold, and T. Sikora. Splitting gaussians in mixture models. In *Proc. IEEE Int. Conf. on Advanced Video and Signal-based Surveillance*, 2012.

48. R. Evangelio and T. Sikora. Complementary background models for the detection of static and moving objects in crowded environments. In *Proc. IEEE Int. Conf. on Advanced Video and Signal-based Surveillance*, 2011.

50. N. Friedman and S. Russell. Image segmentation in video sequences: a probabilistic approach. In *Proc of Uncertainty in artificial intelligence*, pages 175–181, 1997.

52. X. Gao, T. Boult, F. Coetzee, and V. Ramesh. Error analysis of background adaption. *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2000.

54. N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. changedetection.net: A new change detection benchmark dataset. In *IEEE Workshop on Change Detection*, 2012.

56. N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. changedetection.net: A new change detection benchmark dataset. In *Change Detection Workshop (CDW)*, Providence, Rhode Island, June 2012.

58. T. Haines and T. Xiang. Background subtraction with dirichlet processes. In *Proc. European Conf. Computer Vision*, pages 97–111, 2012.

60. B. Han and L. Davis. Density-based multifeature background subtraction with support vector machine. *IEEE Trans. Pattern Anal. Machine Intell.*, 34(5), 2012.

62. Z. Hao, W. Wen, Z. Liu, and X. Yang. Real-time foreground-background segmentation using adaptive support vector machine algorithm. In *Proc of ICANN*, pages 603–610, 2007.

64. M. Hofmann. Background segmentation with feedback: The pixel-based adaptive segmenter. In *IEEE Workshop on Change Detection*, 2012.

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

68. O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *Workshop on Motion and Video Computing*, pages 22–27, December 2002.

70. J-P. Jodoin, G. Bilodeau, and N. Saunier. Background subtraction based on local shape. Technical Report arXiv:1204.6326v1, Ecole Polytechnique de Montreal, 2012.

72. P-M. Jodoin, M. Mignotte, and J. Konrad. Statistical background subtraction methods using spatial cues. *IEEE Trans. Circuits Syst. Video Technol.*, 17(12):1758–1763, 2007.

74. P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for realtime tracking with shadow detection. *European Workshop on Advanced Video Based Surveillance Systems*, 2001.

76. M. Karaman, L. Goldmann, D. Yu, and T. Sikora. Comparison of static background segmentation methods. In *Proc. SPIE Visual Communications and Image Process.*, 2005.

78. K. Karman and A. von Brandt. Moving object recognition using an adaptive background

memory. In Capellini, editor, *Time-varying Image Processing and Moving Object Recognition*, volume II, pages 297–307, Amsterdam, The Netherlands, 1990. Elsevier.

80. H. Kim, R. Sakamoto, I. Kitahara, T. Toriyama, and K. Kogure. Robust foreground extraction technique using gaussian family model and multiple thresholds. In *ACCV*, pages 758–768, 2007.
82. K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model, real-time imaging. *Real-Time Imaging*, 11(3):172–185, 2005.
84. D. Kit, B. Sullivan, and D. Ballard. Novelty detection using growing neural gas for visuo-spatial memory. In *IEEE IROS*, 2011.
86. J. Kittler, M. Hatef, and R. Duin. Combining classifiers. In *IEEE International Conference on Pattern Recognition (ICPR)*, volume 2, pages 897–901, Vienna, Austria, August 1996.
88. F. Kristensen, P. Nilsson, and V. Öwall. Background segmentation beyond rgb. In *Proc of ACCV*, pages 602–612, 2006.
90. L. Kuncheva, C. Whitaker, C. Shipp, and R. Duin. Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis & Applications*, 6(1):22–31, April 2003.
92. T. Landgrebe, P. Paclík, R. Duin, and A. Bradley. Precision-recall operating characteristic (P-ROC) curves in imprecise environments. In *IEEE International Conference on Pattern Recognition (ICPR)*, volume 4, pages 123–127, Hong Kong, August 2006.
94. L. Li, W. Huang, I. Y. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans. Image Process.*, 13(11):1459–1472, 2004.
96. Y. Li. On incremental and robust subspace learning. *Pattern Recognit.*, 37:1509–1518, 2004.
98. C-C Lien, Y-M Jiang, and L-G Jang. Large area video surveillance system with handoff scheme among multiple cameras. In *MVA*, pages 463–466, 2009.
100. H.-H. Lin, T.-L. Liu, and J.-H. Chuang. A probabilistic SVM approach for background scene initialization. In *Proc. IEEE Int. Conf. Image Processing*, pages 893–896, 2002.
102. X. Lu and R. Manduchi. Fast image motion segmentation for surveillance applications. *Image Vis. Comput.*, 29(2-3):104–116, 2011.
104. L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168–1177, 2008.
106. L. Maddalena and A. Petrosino. The SOBS algorithm: what are the limits? In *IEEE Workshop on Change Detection*, 2012.
108. A. Manzanera. Local jet feature space framework for image processing and representation. In *International Conference on Signal Image Technology & Internet-Based Systems*, pages 261–268, Dijon, France, 2011.
110. N. McFarlane and C. Schofield. Segmentation and tracking of piglets in images. *Mach. vis. and appl.*, 8:187–193, 1995.
112. J.M. McHugh, J. Konrad, V. Saligrama, and P.-M. Jodoin. Foreground-adaptive background subtraction. *IEEE Signal Process. Lett.*, 16(5):390–393, May 2009.
114. E. Memin and P. Perez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Trans. Image Process.*, 7(5):703–719, 1998.
116. A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pages 302–309, 2004.

118. A. Morde, X. Ma, and S. Guler. Learning a background model for change detection. In *IEEE Workshop on Change Detection*, 2012.
120. J. Nascimento and J. Marques. Performance evaluation of object detection algorithms for video surveillance. *IEEE Trans. Multimedia*, 8(8):761–774, 2006.
122. Y. Nonaka, A. Shimada, H. Nagahara, and R. Taniguchi. Evaluation report of integrated background modeling based on spatio-temporal features. In *IEEE Workshop on Change Detection*, 2012.
124. N. M. Oliver, B. Rosario, and A. P. Pentland. A Bayesian computer vision system for modeling human interactions. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(8):831–843, 2000.
126. S. K. Mitra P. Jaikumar, A. Singh. Background subtraction in videos using bayesian learning with motion information. In *British Mach. Vis. Conf*, pages 1–10, 2008.
128. T. Parag, A. Elgammal, and A. Mittal. A framework for feature selection for background subtraction. In *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pages 1916–1923, New York, USA, June 2006.
130. J. Park, A. Tabb, and A. Kak. Hierarchical data structure for real-time background subtraction. In *IEEE International Conference on Image Processing (ICIP)*, pages 1849–1852, 2006.
132. D. Parks and S. Fels. Evaluation of background subtraction algorithms with post-processing. In *Proc. IEEE Int. Conf. on Advanced Video and Signal-based Surveillance*, pages 192–199, September 2008.
134. M. Piccardi. Background subtraction techniques: a review. In *Conf. IEEE Int. Conf. Sys., Man and Cyb.*, pages 3099–3104, 2004.
136. F. Porikli. Multiplicative background-foreground estimation under uncontrolled illumination using intrinsic images. In *Proc. of IEEE Motion Multi-Workshop,* Breckenridge, 2005.
138. F. Porikli and O. Tuzel. Bayesian background modeling for foreground detection. *Proc. of ACM Visual Surveillance and Sensor Network*, 2005.
140. F. Porikli and C. Wren. Change detection by frequency decomposition: wave-back. *Proc. of Workshop on Image Analysis for Multimedia Interactive Services,* Montreux, 2005.
142. A. Prati, R. Cucchiara, I. Mikic, and M. Trivedi. Analysis and detection of shadows in video streams: A comparative evaluation. In *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pages 571–577, 2001.
144. R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Trans. Image Process.*, 14:294–307, 2005.
146. D. Riahi, P. St-Onge, and G. Bilodeau. RECTGAUSS-tex: Block-based background subtraction. Technical Report EPM-RT-2012-03, Ecole Polytechnique de Montreal, 2012.
148. P. Rosin and E. Ioannidis. Evaluation of global image thresholding for change detection. *Pattern Recognit. Lett.*, 24:2345–2356, 2003.
150. J. Rymel, J. Renno, D. Greenhill, J. Orwell, and G.A. Jones. Adaptive eigen-backgrounds for object detection. In *Proc. IEEE Int. Conf. Image Processing*, pages 1847 – 1850, 2004.
152. A. Schick, M. Bäuml, and R. Stiefelhagen. Improving foreground segmentations with probabilistic superpixel markov random fields. In *IEEE Workshop on Change Detection*, 2012.
154. M. Sedky, M. Moniri, and C. Chibelushi. Object segmentation using full-spectrum matching of albedo derived from colour images, Patent application PCT GB2009/002829, 2009.
156. C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking.

IEEE Trans. Pattern Anal. Machine Intell., 22(8):747–757, 2000.

158. B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. Buhmann. Topology free Hidden Markov Models: application to background modeling. *Proc. IEEE Int. Conf. Computer Vision*, 2001.
160. J. Suhr, H. Jung, G. Li, and J. Kim. Mixture of gaussians-based background subtraction for Bayer-pattern image sequences. *IEEE Trans. Circuits Syst. Video Technol.*, 21(3):365–370, March 2011.
162. D. Tax, M. van Breukelen, R. Duin, and J. Kittler. Combining multiple classifiers by averaging or by multiplying? *Pattern Recognition*, 33(9):1475–1485, 2000.
164. Y-L. Tian and A. Hampapur. Robust salient motion detection with complex background for real-time video surveillance. In *WACV/MOTION*, pages 30–35, 2005.
166. F. Tiburzi, M. Escudero, J. Bescos, and J. Martinez. A ground truth for motion-based video-object segmentation. In *Proc. IEEE Int. Conf. Image Processing*, pages 17–20, 2008.
168. K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Proc. IEEE Int. Conf. Computer Vision*, volume 1, pages 255–261, 1999.
170. M. Van Droogenbroeck and O. Paquot. Background subtraction: Experiments and improvements for ViBe. In *IEEE Workshop on Change Detection*, pages 32–37, Providence, Rhode Island, USA, June 2012.
172. R. Vezzani and R. Cucchiara. Video surveillance online repository (ViSOR): an integrated framework. *Multimedia Tools and Applications*, 50(2):359–380, 2010.
174. H. Wang and D. Suter. A consensus-based method for tracking: Modelling background scenario and foreground appearance. *Pattern Recognit.*, 40(3):1091–1105, March 2007.
176. W-H. Wang and R-C. Wu. Fusion of luma and chroma gmms for hmm-based object detection. In *Proc of. conference on Advances in Image and Video Technology*, pages 573–581, 2006.
178. S. Warfield, K. Zou, and W. Wells. Simultaneous truth and performance level estimation (STAPLE): An algorithm for the validation of image segmentation. *IEEE Transactions on Medical Imaging*, 23(7):903–921, July 2004.
180. L. Wixson. Detecting salient motion by accumulating directionally-consistent flow. *IEEE Trans. Pattern Anal. Machine Intell.*, 22, 2000.
182. C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(7):780–785, 1997.
184. L. Xu, A. Krzyżak, and C. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):418–435, May-June 1992.
186. M. Xu and T. Ellis. Illumination-invariant motion detection using colour mixture models. In *British Mach. Vis. Conf*, pages 163–172, 2001.
188. Z. Xu, P. Shi, and I. Yu-Hua Gu. An eigenbackground subtraction method using recursive error compensation. In *PCM*, pages 779–787, 2006.
190. S. Yoshinaga, A. Shimada, and Taniguchi R. Statistical background model considering relationships between pixels. In *IEEE Workshop on Change Detection*, 2012.
192. D. Young and J. Ferryman. PETS metrics: Online performance evaluation service. In *Proc. IEEE Int. Workshop on Performance Evaluation of Tracking Systems*, pages 317–324, 2005.
194. C. Zhao, X. Wang, and W-K. Cham. Background subtraction via robust dictionary learning. *EURASIP Journal on Image and Video Processing*, 2011.
196. J. Zheng, Y. Wang, N. Nihan, and E. Hallenbeck. Extracting roadway background

image: A mode based approach. *J. of Transp. Research Report*, pages 82–88, 2006.

198. J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *Proc. IEEE Int. Conf. Computer Vision*, pages 44–50, 2003.

200. D. Zhou and H. Zhang. Modified GMM background modeling and optical flow for detection of moving objects. In *Conf. IEEE Int. Conf. Sys., Man and Cyb.*, pages 2224–2229, 2005.

202. Z. Zivkovic and F. Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognit. Lett.*, 27:773–780, 2006.