Real-time nonparametric background subtraction with tracking-based foreground update

Daniel Berjón[a], Carlos Cuevas[a], Francisco Morán[a], Narciso García[a]

[a] *Grupo de Tratamiento de Imágenes (GTI), Information Processing and Telecommunications Center (IPTC) and ETSI Telecomunicación. Universidad Politécnica de Madrid (UPM), 28040 Madrid, Spain,*

Abstract

A nonparametric real-time and high-quality moving object detection strategy in a GPU is proposed. To improve the quality of the results in sequences where the moving objects and the background have similar appearance, not only the background but also the foreground is modelled. Both models are constructed from spatio-temporal reference data to reduce false detections due to small displacements of the background, and to take into consideration the natural displacements of the foreground. To avoid using kernels with too large spatial widths, the spatial positions of the foreground reference data are updated at each new frame using a particle filter that is able to deal with an unknown and variable amount of regions. Additionally, an automatic selection of regions of interest is carried out, which allows reducing drastically the computational cost of both foreground and background models.

The proposed strategy has been validated using three databases containing many challenges for motion detection and the results have been compared to those of other state-of-the-art approaches.

Keywords: foreground segmentation, background subtraction, nonparametric modelling, parallel processing, real-time GPU

1. Introduction

Foreground segmentation is a key building block of many computer vision algorithms [21]. In general, users of foreground strategies demand robustness and quality results in a wide variety of scenes, as well as real-time performance.

Common methods, such as Gaussian Mixture Model (GMM) [7], can work in real time but their results are very dependent on adequate parameter settings for each scene and do not work well in common situations such as dynamic backgrounds, illumination changes, or bootstrapping sequences (where moving

Email addresses: dbd@gti.ssr.upm.es (Daniel Berjón), ccr@gti.ssr.upm.es (Carlos Cuevas), fmb@gti.ssr.upm.es (Francisco Morán), narciso@gti.ssr.upm.es (Narciso García)

objects are present from the very beginning of the scene) [4]. In addition, only the background is usually modelled; modelling the foreground is not only more computationally demanding but also more challenging because foreground objects are in motion and therefore current pixels must be compared with past pixels that do not lie in the same position but in other positions that need to be determined.

Alternatives to classical modelling based on nonparametric modelling (NPM), and more specifically based on kernel density estimation (KDE), have been proposed [39]. Some recent works also describe ways to efficiently implement such strategies using GPUs to achieve real-time operation. This paper presents a comprehensive detection system firmly rooted upon KDE-based foreground and background modelling, augmented with auxiliary tracking and selective analysis modules, that produces higher quality detections than previous proposals.

The specific contributions of this paper are: 1) The background model is combined with a foreground model, both of which use spatio-temporal reference information; the foreground model can update the positions of reference data using a particle filter that is able to manage a variable and a priori unknown number of moving regions; 2) A Bayesian classifier that is able to combine models with different spatial widths by conditioning them by their spatial marginal distributions. Thus, it is possible to simultaneously employ small spatial widths in the background model to reduce noise and larger spatial widths in the foreground model to cope with the inherent imprecisions of motion estimation of the reference data; 3) A selective analysis strategy based on random sampling and regions of interest (RoI) that yields results comparable to those of a full analysis at a fraction of the cost; 4) A computationally efficient method for dynamically selecting the appropriate appearance kernel width for the background model from a fixed set of values ; 5) A practical, GPU-based, real-time implementation of all the proposed features.

Extensive evaluation of the proposed approach using several publicly available databases that feature a variety of typical challenges (dynamic backgrounds, illumination changes, camouflage, low contrast, bootstrapping, etc.) has been carried out to demonstrate its performance and compared against other state-of-the-art alternatives.

All the results reported in this paper, along with the complete source code of the proposed system, are available to the public[1] so that other researchers can replicate our results, use our system with any other database or compare it to their own proposals.

2. Related work

A large amount of moving object detection approaches has been proposed in the literature over recent years [6, 13, 42]. Depending on the complexity of the problem, some algorithms use pixel features [17], analyze the motion in the scene

[1] www.gti.ssr.upm.es/data/

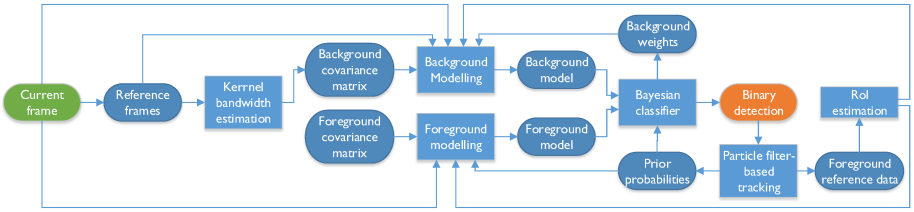

Figure 1: Block diagram of the proposed system. Notation: round-edged blocks denote data and rectangular blocks denote processes. The green and orange blocks indicate, respectively, the input and the output of the system.

[46], combine colour with motion information [10] or make use of the depth data of the scene [36]. Some strategies aim to maximize the computational efficiency and to reduce the computational requirements. However, they only provide successful detections in short sequences with quasi-stationary background [38]. Other strategies employ spatial [37] or spatio-temporal [27] features to improve the detections in noisy sequences. However, they fail in complex situations, such as scenarios with dynamic background or illumination changes [28].

To improve the quality of the results in such situations, several multimodal strategies have been proposed, which are able to model multiple states for each pixel [6]. Therefore, they adequately deal with sequences with dynamic background areas [20] containing elements with cyclic movements (e.g. rain, flags, trees, sea waves, etc.). Among the multimodal methods, the GMM proposed by [43] must be highlighted, since it has been taken as starting point by hundreds of authors (see [7]) and it has thousands of citations. GMM-based methods employ a few Gaussians (typically 3 to 5) to obtain an adaptive model of each image pixel, which allows them to provide high-quality detections in many complex scenarios [45]. However, they fail in environments where the pixel level can not be described parametrically [39]. Additionally, they depend on many parameters that must be manually changed according to the characteristics of the sequence to analyze, thereby decreasing their usability [12].

To solve these drawbacks, nonparametric kernel density estimation (KDE) approaches have also been developed. Instead of trying to fit the observations to a preconceived probability distribution, the actual distribution is estimated from the recent history of each pixel and its neighbours as a superposition of kernels, typically Gaussian [18]. In KDE approaches, unlike in GMM-based ones, the number of parameters to be manually selected according to the characteristics of the sequences is very low. Actually, only the width of the kernels used to construct the model must be manually set. However, some algorithms to automatically estimate these widths have been proposed [5, 23] and, consequently, these detection methods are very easy to use. Although KDE detection algorithms provide very high quality in complex and multimodal scenarios and have a very high usability, their main drawback is their extremely high computational and memory cost: a comprehensive history of every pixel is stored as reference to directly (i.e., non recursively) estimate and evaluate a probability

density function (pdf) per each new input pixel and channel. As a result of these considerable demands, it has received comparatively little attention in the literature because it is impracticable in real-time scenarios without resorting to massive parallelisation using GPUs [3, 11].

3. System overview

The main blocks of the proposed detection system and their interconnections are shown in Fig. 1.

The core of the system is a KDE-based modelling strategy, whose workings can be summarized as follows: for each frame in an input sequence, the pdfs that each image pixel belongs to the image background or foreground are non-parametrically estimated using spatio-temporal KDE as described, respectively, in sections 4.1 and 4.2. These models are introduced in a novel Bayesian classifier that can combine pdfs obtained with spatio-temporal kernels with different spatial width. The result provided by this classifier, detailed in section 4.3, is a mask determining what pixels belong to the moving objects in the image. While this modelling strategy is very adequate to model complex and changing scenes and yields good foreground segmentations, it has the severe drawback of a high computational cost. Therefore, the rest of the modules of the system are ancillary to the main KDE model and aim at improving the quality of the detections while also cutting down on computational cost.

The foreground mask from the classifier is fed to a tracking module. In this module, described in section 5, a new tracking strategy based on the simultaneous application of multiple particle filters is applied, which allows dealing with an unknown and variable number of moving regions while maintaining an approximately constant computational cost. The data resulting from the region tracking is used to update the spatial coordinates of the reference foreground data, which improves the quality of the foreground modelling and allows using kernels with small spatial width. Additionally, the filters provide prior information that is fed back into the Bayesian classifier to enhance discrimination between foreground and background. Finally, taking into account the foreground mask obtained for the current image and the tracking results, a region of interest is predicted for the next frame, as described in section 6, resulting in a drastic reduction of the computational cost in the estimation of both background and foreground models. The combination of all the modules, together with massive parallelisation in a GPU, makes the proposed system apt to be used in scenarios demanding real-time operation, e.g. surveillance, traffic analysis, etc.

4. Spatio-temporal nonparametric modelling

4.1. Background modelling

Let p^n be a pixel in the current image I^n at time n. Let such pixel be defined by a $(D{+}2)$-dimensional vector, $\mathbf{x}^n = ((\mathbf{a}^n)^T, (\mathbf{s}^n)^T)^T$, where $\mathbf{a}^n$ is a

D-dimensional vector containing the appearance information of the pixel and $\mathbf{s}^n = (r^n, c^n)$ is a vector containing its spatial coordinates (row and column). Let $\{\mathbf{x}_\beta^i\}_{i=1}^{N_\beta}$ be a set of N_β $(D+2)$-dimensional reference samples, obtained from T_β previous images into a spatial neighbourhood arround (r^n, c^n). Applying $(D+2)$-variate Gaussian kernels, the pdf that p^n belongs to the image background, β, is estimated as

$$\mathrm{p}(\mathbf{x}^n|\beta) = \sum_{i=1}^{N_\beta} w_i N\left(\mathbf{x}^n - \mathbf{x}_\beta^i; \Sigma_{\beta,\mathbf{x}^n}\right) \tag{1}$$

where $\Sigma_{\beta,\mathbf{x}^n} = \mathrm{diag}(\sigma^2_{\beta,\mathbf{x}^n,1}, \sigma^2_{\beta,\mathbf{x}^n,2}, \ldots, \sigma^2_{\beta,\mathbf{x}^n,D}, \sigma^2_{\beta,\mathbf{x}^n,R}, \sigma^2_{\beta,\mathbf{x}^n,C})$ is a diagonal covariance matrix that determines the width of the kernels and $\{w_i\}$ are weighting coefficients (obtained as in [12]) that sum up to one and allow to selectively update the model. Unlike parametric fitting of a mixture of Gaussians, this kernel density estimation is a more general approach that does not assume any specific shape for the density function.

To prevent the evaluation of data not contributing significantly to this estimation, the spatial neighbourhood is limited to samples satisfying $D_{Mah}\left(\mathbf{s}^n, \mathbf{s}_\beta^i\right) \leq d_{\max}$, where $D_{Mah}\left(\mathbf{s}^n, \mathbf{s}_\beta^i\right)$ is the Mahalanobis distance between the spatial coordinates of the current sample and those of the i-th reference sample using $\Sigma_{\beta,\mathbf{s}^n} = \mathrm{diag}(\sigma^2_{\beta,\mathbf{x}^n,R}, \sigma^2_{\beta,\mathbf{x}^n,C})$ as covariance matrix, and d_{max} is set to 3.44 to guarantee that only those reference samples falling inside the 99.7% of the spatial Gaussian kernels defined by $\sigma^2_{\beta,\mathbf{x}^n,R}$ and $\sigma^2_{\beta,\mathbf{x}^n,C}$ are considered [19].

The reference samples are uniformly distributed in space. Thus, the same fixed spatial widths can be used in the whole image extent. The values assigned to these parameters as well as the justification for such assignment appear in section 7.

4.1.1. Dynamic Width Switching

The kernel widths corresponding to the appearance components of the background are dynamically estimated as

$$\sigma_{\beta,\mathbf{x}^n,j} = \sum_{r,c} w_\Sigma\left(r,c\right) S^n\left(r,c,j\right) : j \in [1, D]\,, \tag{2}$$

where $w_\Sigma\left(r,c\right)$ is a normalized weighting factor, inversely related to the spatial distance between samples, that is obtained as

$$w_\Sigma\left(r,c\right) \propto \exp\left(-\frac{1}{2} D_{Mah}\left(\mathbf{s}^n, \mathbf{s}_\beta^i\right)\right), \tag{3}$$

and $S^n\left(r,c,j\right)$ is an initial width computed as

$$S^n\left(r,c,j\right) = \frac{m\left(r,c,j\right)}{0.68\sqrt{2}}, \tag{4}$$

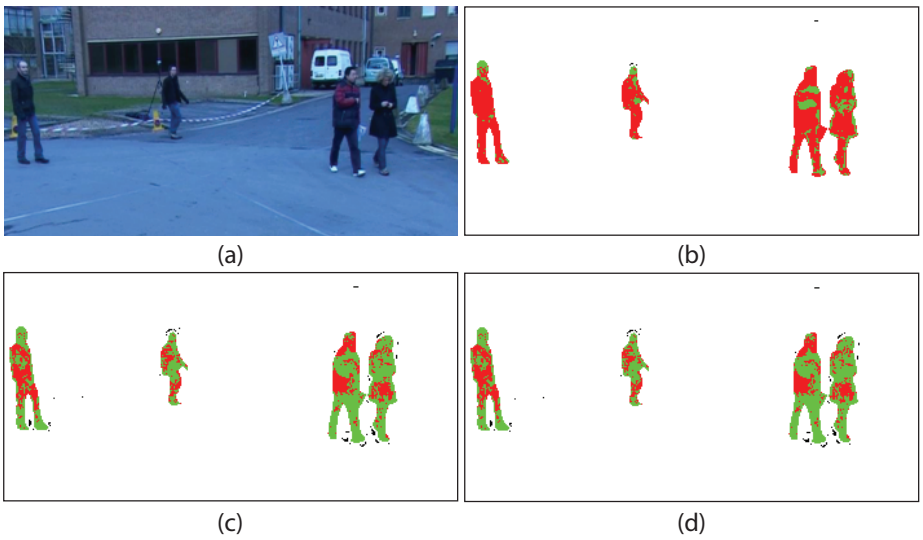

Figure 2: Detections in an outdoor sequence using different appearance width estimators: (a) Original image; (b) detection using the weighted mean from [3]; (c) detection using the median; (d) detection using the proposed width switching scheme. Colour notation (for this figure and all subsequent ones showing detection results): correct detections (green), misdetections (red), and false detections (black).

where $m\left(r,c,j\right)$ is the median of the absolute value of the set of differences between the j-th appeareance component of consecutive reference samples at the coordinates (r,c), $\left\{\mathbf{a}_{\beta}^{i}\left(r,c,j\right)-\mathbf{a}_{\beta}^{i-1}\left(r,c,j\right)\right\}_{i=n-T_{\beta}}^{n-1}$ [18].

Medians are fairly expensive to compute (between $O(\log T_{\beta})$ and $O(T_{\beta})$ depending on the data structures available). A more efficient ($O(1)$) estimator is a moving average, weighted with past detections to discard outliers [3]. Both methods usually yield equivalent results, but misdetections tend to upset the estimation in the latter case, as illustrated in Fig. 2; therefore, we propose a robust and efficient estimator based on per-channel log-spaced histograms.

Absolute differences between consecutive samples with normalized data lie in the range $[0,1]$, so we partition this interval in bins defined by the knots $\{0\}\cup\{\exp(\log(\sigma_{min})(1-\frac{k-1}{N_{\sigma}-1}))\}_{k=1}^{N_{\sigma}}$, where σ_{min} is the minimum meaningful difference in the colour space that is being used and N_{σ} is the number of bins of the histogram. In each frame, one bin is incremented and another decremented with cost $O(1)$, then the cumulative sum is computed ($O(N_{\sigma})$, independent of T_{β}) to find the bin b where the true median is contained. The greater N_{σ}, the more this estimator converges to the median. Finally, the median is estimated as

$$m\approx\begin{cases}\sigma_{min} & b=1\\ \exp(\log(\sigma_{min})(1-\frac{2b_{j}-3}{2(N_{\sigma}-1)})) & b\in\{2,...,N_{\sigma}\}\end{cases}.$$

As we discuss in section 7, even for low N_σ the obtained results are virtually undistinguishable from using the median (compare Fig. 2.c and Fig. 2.d).

4.2. Foreground modelling

The pdf that p^n belongs to the image foreground, ϕ, can be non-parametrically estimated [39] as a mixture of a constant density, γ, of a uniform random variable in the $D+2$ components defined for the feature vector $\mathbf{x}^n$ and a density function estimated with $(D+2)$-variate Gaussian kernels:

$$\mathrm{p}\left(\mathbf{x}^n|\phi\right) = \alpha\gamma + \frac{1-\alpha}{N_\phi}\sum_{i=1}^{N_\phi} N\left(\mathbf{x}^n - \mathbf{x}_\phi^i; \Sigma_{\phi,\mathbf{x}^n}\right) \tag{5}$$

where α is a mixture factor, $\{\mathbf{x}_\phi^i\}_{i=1}^{N_\phi}$ is the set of $(D+2)$-dimensional foreground samples stored along the last T_ϕ images into a spatial neighbourhood around the coordinates of p^n, and $\Sigma_{\phi,\mathbf{x}^n} = \mathrm{diag}(\sigma^2_{\phi,\mathbf{x}^n,1}, \sigma^2_{\phi,\mathbf{x}^n,2} \cdots \sigma^2_{\phi,\mathbf{x}^n,D}, \sigma^2_{\phi,\mathbf{x}^n,R}, \sigma^2_{\phi,\mathbf{x}^n,C})$ is the covariance matrix determining the width of the kernels. Just as in the background modelling, to avoid the evaluation of reference samples not contributing significantly to the estimation process, only samples satisfying $D_{Mah}(\mathbf{s}^n, \mathbf{s}_\phi^i) \leq d_{max} = 3.44$, with $\Sigma_{\phi,\mathbf{s}^n} = \mathrm{diag}(\sigma^2_{\phi,H}, \sigma^2_{\phi,W})$ as covariance matrix, are considered.

The spatial width values used in this foreground modulating depend on the number of reference images, T_ϕ, and on the speed of the moving objects, i.e. they should be large enough to take into account all the reference data in all the reference images [11]. Therefore, in principle, the spatial widths used in the foreground should be significantly larger than those used in the background, which results in a high computational cost. As discussed in section 5, the application of the proposed tracking strategy allows to reduce these widths and make them independent of the speed of the objects and the amount of reference images.

Appearance widths cannot be determined using the same procedure of the background because the distribution of reference data is not dense or regular enough. In addition, variance is not only caused by capture noise but also by the changes in the observer's angle of view. Therefore, we resort to manually setting this parameter, which is relatively insensitive in most sequences. It must, however, be set higher than the appearance width of the background so that false detections do not feed back and become persistent.

4.3. Conditioned Bayesian classifier

As said before, on the one hand, the spatial width used in the background modelling must be small. However, on the other hand, the width applied to the foreground modelling must be large enough to cover the object displacements along the reference images. Consequently, it is desirable to use different width values in each model. However, foreground and background reference data are distributed in space very differently, which sometimes results in significant mismatch in the magnitudes of the spatial distributions of the data ($\mathrm{p}(\mathbf{s}^n|\phi)$ and

$\mathrm{p}(\mathbf{s}^n|\beta)$) in some regions, leading to persistent false detections. Therefore, instead of the typical Bayesian classifier [31], we propose an alternative to decouple the appearance and spatial information of both models:

$$\Pr\left(\phi|\mathbf{x}^n\right) = \frac{\Pr\left(\phi|\mathbf{s}^n\right)\mathrm{p}\left(\mathbf{a}^n|\phi,\mathbf{s}^n\right)}{\Pr\left(\phi|\mathbf{s}^n\right)\mathrm{p}\left(\mathbf{a}^n|\phi,\mathbf{s}^n\right) + \Pr\left(\beta|\mathbf{s}^n\right)\mathrm{p}\left(\mathbf{a}^n|\beta,\mathbf{s}^n\right)} \tag{6}$$

where $\Pr\left(\phi|\mathbf{s}^n\right)$ and $\Pr\left(\beta|\mathbf{s}^n\right) = 1 - \Pr\left(\phi|\mathbf{s}^n\right)$ are, respectively, the foreground and background prior probabilities (obtained from the results provided by the tracking strategy as described in section 5); $\mathrm{p}\left(\mathbf{a}^n|\phi,\mathbf{s}^n\right)$ and $\mathrm{p}\left(\mathbf{a}^n|\beta,\mathbf{s}^n\right)$ result from conditioning the foreground and background models, $\mathrm{p}\left(\mathbf{x}^n|\phi\right)$ and $\mathrm{p}\left(\mathbf{x}^n|\beta\right)$, on a particular spatial location. These conditioned density functions are obtained as

$$\mathrm{p}\left(\mathbf{a}^n|\xi,\mathbf{s}^n\right) = \frac{\mathrm{p}\left(\mathbf{x}^n|\xi\right)}{\mathrm{p}\left(\mathbf{s}^n|\xi\right)} : \xi \in \{\beta,\phi\} \tag{7}$$

where $\mathrm{p}\left(\mathbf{s}^n|\xi\right)$ is the marginalization of $\mathrm{p}\left(\mathbf{x}^n|\xi\right)$ over the D-dimensional set of appearance characteristics. These marginal density functions are obtained as

$$\mathrm{p}(\mathbf{s}^n|\beta) = \sum_{i=1}^{N_\beta} w_i N\left(\mathbf{s}^n - \mathbf{s}_\beta^i; \Sigma_{\beta,\mathbf{s}^n}\right) \tag{8}$$

$$\mathrm{p}\left(\mathbf{s}^n|\phi\right) = \alpha\gamma' + \frac{1-\alpha}{N_\phi}\sum_{i=1}^{N_\phi} N\left(\mathbf{s}^n - \mathbf{s}_\phi^i; \Sigma_{\phi,\mathbf{s}^n}\right) \tag{9}$$

where γ' is a constant density in the spatial components. These densities are obtained without further computational effort (cf. eqs. (1) and (8), or (5) and (9)). Fig. 3 illustrates some results obtained with the typical Bayesian classifier and with the proposed classifier. Although moving objects are correctly detected in both cases, the typical classifier yields many persistent false detections, whereas the proposed one mitigates them.

5. Object tracking with a particle filter

The model of the foreground uses as reference data only those pixels that were classified as foreground in past frames instead of the whole images as proposed in [39]. However, the foreground is made up, by definition, of moving objects, and this poses a problem: if we look for reference pixels in a small spatial vicinity of each current pixel in recent frames, we will find only a few reference pixels unless the object is moving quite slowly; this in turn means that the foreground model will be very noisy and unreliable.

One obvious solution to this problem is increasing the size of the region where reference pixels will be sought. Unfortunately, this is computationally expensive [11] and, moreover, it may result in using pixels from different objects as

Figure 3: (a) Original images from two video sequences. (b) Detections using the typical Bayesian classifier. (c) Detections using the proposed conditioned classifier.

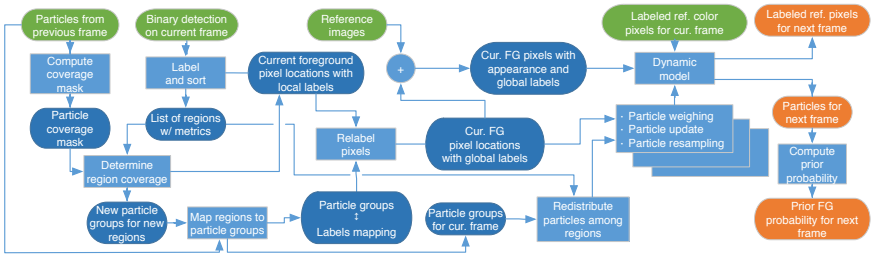

Figure 4: Block diagram of the proposed tracking system. Round-edged blocks denote data (green is input; orange is output) and rectangular blocks denote processes.

reference, which decreases the quality of the resulting model. A better solution to the problem is to model the motion of each object in the foreground so that reference data from recent frames can be projected onto the estimated location of each object in the current (to be processed) frame. This both allows to reduce the search area for reference data and increases the probability that the data used to model each pixel is actually relevant to it. Consequently, we propose a multi-object tracking method based on a particle filter framework that is able to deal with an unknown and variable number of moving regions throughout the sequence. Many methods have been proposed for multiple object tracking, but most of them assume object detection has already been performed and work at the object level [34, 41]. Our proposal works instead at a lower level (i.e., connected regions of pixels) and is object-agnostic because its only aim is to compact reference data for the foreground model at a low computational cost,

rather than provide long-term object identification.

5.1. General description

The proposed algorithm is based on a particle filter of the Sequential Important Resampling (SIR) type [1] and features all its typical stages [47]: prediction, weights update, normalization and resampling. Our goal is to estimate iteratively the probability density function of a state vector $\mathbf{u}^n$ from a set of measures $\mathbf{v}^n$ obtained from the n-th input image, I^n. This probability density function can be defined as

$$\begin{aligned} &\mathrm{p}\big(\mathbf{u}^n|\mathbf{v}^n, \mathbf{v}^{n-1}, \ldots, \mathbf{v}^1\big) \\ &\propto \mathrm{p}(\mathbf{v}^n|\mathbf{u}^n)\, \mathrm{p}\big(\mathbf{u}^n|\mathbf{v}^{n-1}, \ldots, \mathbf{v}^1\big)\,, \end{aligned} \tag{10}$$

where $\mathrm{p}\big(\mathbf{u}^n|\mathbf{v}^{n-1}, \ldots, \mathbf{v}^1\big)$ is the probability distribution predicted from past observations [24] and $\mathrm{p}(\mathbf{v}^n|\mathbf{u}^n)$ is the likelihood of the state vector $\mathbf{u}^n$ given the set of J_s measures

$$\mathbf{v}^n = \{\mathbf{v}_m^n\}_{m=1}^{J_s} = \left\{(r,c) : \Pr(\phi|\mathbf{x}^n) > \frac{1}{2}\right\}, \tag{11}$$

which are the locations of the pixels that have been classified as foreground on image I^n. In order to obtain this estimation we evaluate a set of N_s particles $\{\mathbf{u}_i^n, \varpi_i^n\}_{i=1}^{N_s}$, where ϖ_i^n is the weight of the i-th particle in the set.

We have chosen to model each moving region as a bidimensional Gaussian distribution whose axes are parallel to the edges of the image, moving with constant velocity from one frame to the next. Thus, each particle will be represented by the state vector defined as

$$\mathbf{u}_i^n = \left(r_i^n, c_i^n, \dot{r}_i^n, \dot{c}_i^n, \sigma_{r,i}^n, \sigma_{c,i}^n\right)^T, \tag{12}$$

where (r_i^n, c_i^n) is the position of the centre of the i-th distribution, $(\dot{r}_i^n, \dot{c}_i^n)$ is its velocity and $\left(\sigma_{r,i}^n, \sigma_{c,i}^n\right)$ are the standard deviations of the distribution in each of its axes.

While particle filters are inherently multimodal, following multiple objects requires significant adaptations of the basic structure of the filter; Fig. 4 shows the block structure of the proposed multi-region tracking algorithm, that comprises stages that have global scope as well as stages that are replicated in a per region basis to better model the independent behaviour of each tracked object. Throughout the following sections, each of the blocks that make up the proposed tracking system are detailed and referred to by their name in the figure.

5.2. Multiple object management

For each input image I^n, the N_s particles that the filter uses are distributed among as many groups as contiguous regions are detected in that image according to their area, so that the most significant objects get more precise estimations; since the total number of particles remains unchanged, the computational cost of the particle filter remains approximately constant. Each particle

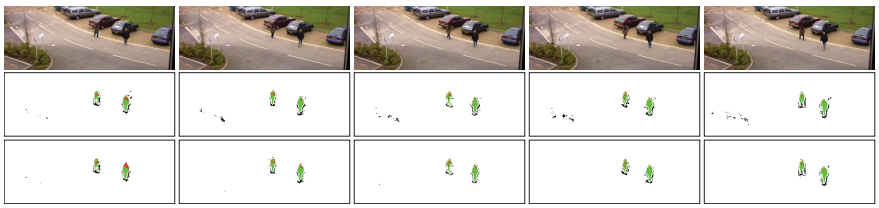

Figure 5: The top row show images from a scene, spaced five frames apart; a gust of wind is blowing the bushes on the left, causing false detections. In the absence of any filtering (middle row), these false detections are fed back into the foreground model, causing persistence. In the bottom row, we show the results of rejecting small regions at the labelling stage. While false detections do occur, they are not fed back into the foreground model, preventing their growth and improving overall detection.

group receives a unique identifier upon creation and lives for as long as it can be reliably identified with the same moving object over time; for this task, it is useful to represent each group by a metaparticle, which is a weighted sum (more details in section 5.3) of all the particles in the group.

5.2.1. Region identification

The *Label and sort* block implements classic connected-components labelling analysis [44] on the set of measures $\mathbf{v}^n$. Regions too small to be meaningful objects are discarded; this removes most false detections from the pool of reference samples, which avoids undesirable feedback leading to persistence (see Fig. 5). The remaining connected components are measured to obtain their area in pixels, their centre and their standard deviation in the vertical and horizontal axes.

5.2.2. Detection of new regions

Among all the regions detected in the *Label and sort* block, some may correspond to objects that were previously detected in the previous frame and some to new objects; we need to determine which (new) regions are not covered by the current set of particles and create new particle groups for them. We consider that a measure is covered by a particle if it is contained within the central region of the distribution associated to the particle that accumulates 99.7% (the common three-sigma rule) of the probability. Therefore, a measure will be covered by a particle if $D_{Mah}(\mathbf{u}_i^n, \mathbf{v}_m^n) \leq 3.44$ [19], where $D_{Mah}(\mathbf{u}_i^n, \mathbf{v}_m^n)$ is the Mahalanobis distance between the measure and the centre of the particle. The *Compute coverage mask* block performs this computation in the GPU for all possible pixel locations in the image to generate a full binary coverage mask.

The *Determine region coverage* block creates new particle groups for new regions; if none of the measures of a region is covered by the existing particle set, a new particle (group) must be created to start tracking that region. The initialization values will be the metrics of the region as computed by the *Label and sort* block.

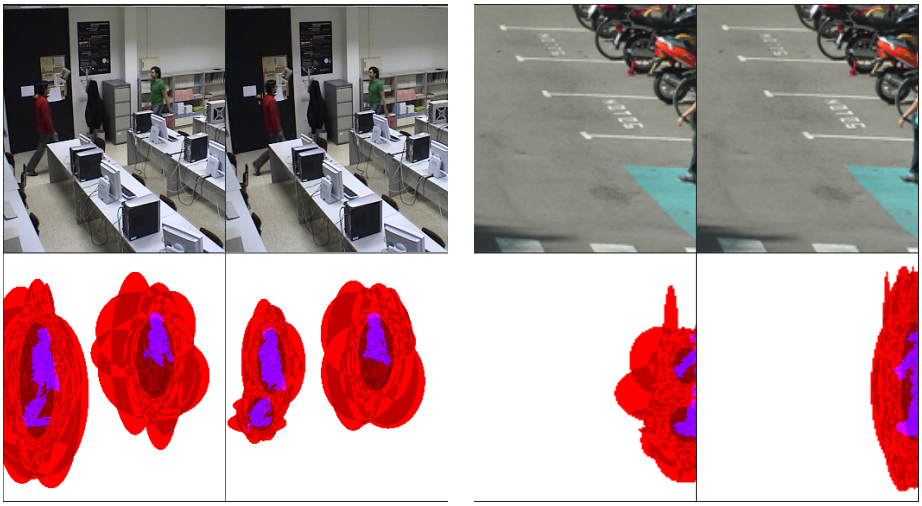

Figure 6: Examples of common occurrences that make the number of existing regions change. Red ellipses depict particles and the darker ellipse surrounding objects shows the representative metaparticle of each particle group. In subfigure (a), the person on the left walks behind the table and his foot cannot be connected to the rest of his body; the group of particles assigned to him splits into two groups. In subfigure (b), a person is entering the frame, but initially his leg and arm are two independent objects that are later recognized as one and their associated particle groups joined.

5.2.3. Disappearance, split and fusion of regions

Now we need to determine the relation between the regions found in the current detection and those found in previous detections to track them correctly. Additionally, we need to take into consideration other common occurrences (see examples in Fig. 6) such as: a) disappearance of previously existing regions (e.g., a moving object exits the frame or is occluded); b) fusion of previously disjoint regions (e.g., several parts of an object are entering the scene, or the projections on the camera of two independent objects overlap, or an object that was partially occluded ceases to be occluded); c) split of a previously unique region (e.g., two objects that were together part ways, or an object gets partially occluded, for instance a person passing behind a lamppost).

As the (admittedly non-exhaustive) enumeration we have just made shows, a single phenomenon may have different explanations that would require different treatments and would require a (costly) high-level analysis. However, our only aim is to compact reference data at least not worse than doing nothing; therefore, we default to treating fused or split regions as new objects; since new particles are created with zero velocity, this is never worse than not using the particle filter.

The *Map regions to particle groups* block implements this policy and yields a bijection between regions in the image and groups of particles. The proposed algorithm is:

- If a particle group (its representative metaparticle) does not cover at all any region, it is immediately deleted.
- We determine which is the preferred particle group for each region (i.e., closest using the Mahalanobis distance for a majority of its measures $\mathbf{v}_m^n$).
- Now we can map each region to its preferred particle group. However, in some cases more than one region can prefer the same group; this indicates that the group is not well adapted to either region. Therefore, we create new groups for all the affected regions.
- Any leftover particle group can simply be deleted.

5.2.4. Redistribution of particles

We want to distribute the configured number of particles among the groups/regions we have found so that every group has at least one particle and the number of particles is proportional to the area of their region. In this operation, a group of particles can vary with respect to the previous frame. If it needs to be downsized, excess particles are selected at random. If the group gains particles, new particles are created with values randomly distributed around the representative metaparticle of the group.

5.3. Region-level tracking

Once all particles have been distributed into groups associated with regions, we perform the typical stages of a particle filter separately on a per particle

group/region basis. Thus, particles will be weighed only in relation with the other particles from their group, resampling will take place inside each group and a different dynamic model will be applied to each group.

Firstly, we need to evaluate how well each particle describes the region it has been assigned to. Let us consider a mobile region $V^n = \{V_\tau^n\}_{\tau=1}^J$, which is made of J measures. The weight of the particles that are associated to this region is computed as

$$\varpi_i^n = L(\mathbf{u}_i^n|V^n) \cdot \varpi_i^{n-1}, \tag{13}$$

where $L(\mathbf{u}_i^n|V^n)$ is a likelihood function that relates the set of measures belonging to the mobile region to each of the particles associated to that region. We have chosen the F-score (harmonic mean of precision and recall) as likelihood function because it does not have any parameters to adjust and produces a well-balanced result.

In this context, we define $J_{in,i}^n$ as the number of measures of the region V^n covered by particle $\mathbf{u}_i^n$. Thus, we can define precision as the fraction of the area covered by a particle that contain measures from the region, $P_i^n = \frac{J_{in,i}^n}{\pi \cdot \sigma_{r,i}^n \cdot \sigma_{c,i}^n \cdot 3.44^2}$; we can define recall as the fraction of measures from the region that are covered by a particle, $R_i^n = \frac{J_{in,i}^n}{J}$. Therefore, the F-score and likelihood function is

$$L(\mathbf{u}_i^n|V^n) = \frac{2P_i^n R_i^n}{P_i^n + R_i^n} = \frac{2J_{in,i}^n}{J + \pi \cdot \sigma_{r,i}^n \cdot \sigma_{c,i}^n \cdot 3.44^2}. \tag{14}$$

Once all the particles in the group have been weighed, we can update the global state estimation of the group as

$$\mathbf{u}^n = \sum_i \varpi_i^n \mathbf{u}_i^n. \tag{15}$$

Weighing all the particles is one of the most computationally demanding stages of the filter and, therefore, we have implemented it in the GPU. Then, we proceed to resample [16] the particles to propagate those with higher weights and eliminate those that did not contribute significantly to the estimation and normalize their weights. Finally, we apply a constant-velocity dynamic model in order to predict their state for the next frame.

5.4. Integration of the results with the nonparametric model

The contribution of the particle filter to the main nonparametric model is twofold: it allows us to displace reference data with the objects it belongs to and, since we are estimating the trajectories and velocities of those objects, we can use this knowledge to establish prior probabilities of each pixel to belong to the foreground. In this subsection we will detail both contributions to the nonparametric model.

Figure 7: Example of prior foreground probability distribution.

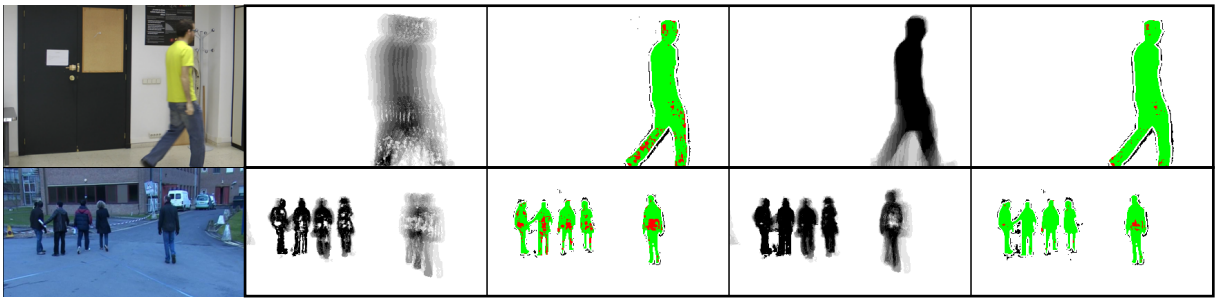

Figure 8: Contributions of the tracking system on two scenes. Column (a) shows the original scenes; (b) shows the foreground reference data distribution when the tracking system is not used; (c) shows results without tracking; (d) shows the foreground reference data distribution when its position is displaced according to the dynamic model estimated by the tracking system and (e) shows final detections with clearly improved recall over (c).

5.4.1. Prior probability estimation

In the absence of any previous information, the prior probability of a location $\mathbf{s}^n$ in the n-th image to belong to the foreground should be established as $1/2$, which is equivalent to not using prior probabilities. However, after applying the dynamic model to all the previously resampled particles, we expect the particles to be concentrated over the areas where the objects are most likely to be located in the next frame. We can use this information to establish a reasonable prior probability in the areas covered by particles, while retaining the default prior probability where there are no particles in order not to hamper the detection of new objects. Thus, we can define

$$\Pr(\phi|\mathbf{s}^n) = \begin{cases} \frac{1}{2}, & N_p = 0 \\ \frac{1}{2} + \frac{1}{2N_p}\sum_{i=1}^{N_p} G_i(\mathbf{s}^n), & N_p > 0 \end{cases} \tag{16}$$

where N_p is the number of predicted particles that cover $\mathbf{s}^n$ and $G_i(\mathbf{s}^n)$ is the contribution of the i-th particle on that position:

$$G_i(\mathbf{s}^n) = \exp\left(-\frac{\left(\mathbf{s}^n(1) - r_i^n\right)^2}{2\left(\sigma_{r,i}^n\right)^2} - \frac{\left(\mathbf{s}^n(2) - c_i^n\right)^2}{2\left(\sigma_{c,i}^n\right)^2} \right). \tag{17}$$

This prior is trivially computed on the GPU; Fig. 7 shows a scene and its map of prior probability of the foreground.

5.4.2. Displacement of reference data

After all the regions in the current frame have been mapped to particle groups, we can replace the local labels we assigned to the measures $\mathbf{v}^n$ with global labels: the identifiers of their corresponding particle group, which they will, in the absence of fusions or splits, share with measures from previous frames (reference data for the next frame) corresponding to the same object; thus, we can displace all the reference data from the same object using the same dynamic model of its associated particle group; Fig. 8 shows clearly improved reference data densities and results when applying the dynamic model to reference data.

6. Selective Analysis and Regions of Interest

Most pixels in typical video sequences belong to the background; therefore, if we can find a way to only analyze those pixels likely to belong to the foreground and classify the rest as background by default we could significantly cut down the computational cost while maintaining the quality of the detections. Since the aim is just to reduce the computational burden, we employ a very simple selective analysis strategy that requires only minimal modifications to the heavily-parallel, GPU-based, core KDE analysis module. The proposed module is based upon these three observations:

Iterative analysis: In most scenarios, relevant moving objects have a minimum size that can be characterized. Therefore, we can divide the input

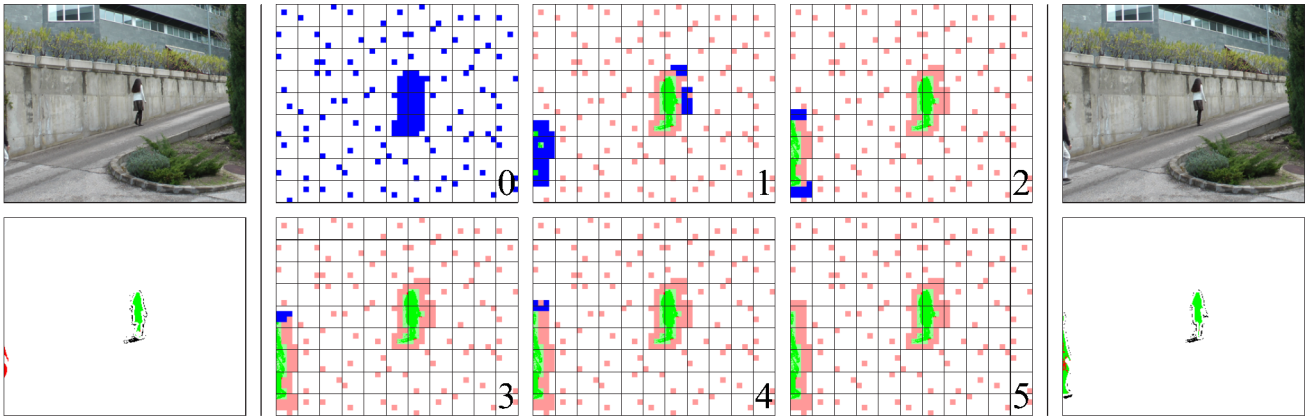

Figure 9: Selective analysis of a frame. Subfigure (a) shows an image featuring an already established object and a new object just entering the frame on the left that has not been hit by WRS, hence the misdetection. Subfigure (b) shows, iteration by iteration, the detection process using blocks of 8×8 px, 4×4 macroblocks for WRS and a growing radius of 2-blocks on the very next image (shown in Subfigure c), where the new object is correctly detected; blue blocks are those to be analyzed in the next iteration and red/green blocks are those that tested negative/positive. We can see in the zeroth iteration that the blocks where the already established object is expected to be located are marked for analysis from the beginning and in the first iteration the new object is hit by WRS, correctly growing to full detection in the next iterations. Subfigure (c) shows the actual image and the final detection.

image into blocks smaller than this minimum size and sample them, maintaining a high probability that the moving objects will be at least partially covered by one or more of the sampled blocks and we will detect them by only applying KDE on these blocks. Then, we can iteratively extend KDE analysis only to those blocks next to those that we have found to contain foreground, guaranteeing full object detection from a single positive seed block.

Windowed Random Sampling (WRS): We would like to reduce the amount of work as much as possible, and we could do that by sampling the blocks very sparsely. One way to look at it is to divide the image into macroblocks of a few blocks per dimension (e.g. 4×4) and select one block per macroblock to analyze. Unfortunately, this systematic sparse sampling carries the risk of whole objects fitting in the space between samples. However, if the temporal sampling rate of the video is high enough in relation with the speed of the apparent motion of objects in the scene, a moving object will typically spend several frames into a macroblock. Therefore, if instead of selecting fixed blocks within each macroblock we randomly choose one or more blocks to analyze, any moving object will be hit in at most a few frames. Selecting random blocks within macroblocks instead of doing it in the image as a whole ensures that no significant portion of the image can be left out.

Temporal coherency: Although the iterative analysis we have proposed guarantees the detection of the full object as soon as one block tests positive, the subsampling should be reasonably dense in order to all objects to be hit in every frame; this in turn means analyzing a significant portion of the image, which is what we want to avoid. However, since we have already designed a tracking system, we designate as RoI the regions where the current detections are projected in the next frame in addition to the blocks sparsely sampled using WRS.

6.1. Implementation

The detailed implementation of the algorithm, illustrated in Fig. 9, is as follows:

1. Input images are tessellated into blocks. Each block can take one of these states: a) *not analyzed* (this is the initial state of every block): this block has not been analyzed and will not be in the next iteration; b) *to be analyzed*: this block will be analyzed in the next iteration of KDE; c) *positive* or *negative*: this block has been analyzed in a previous iteration of KDE and has or not been found to contain a significant amount of foreground.
2. Whenever a new frame arrives, random blocks are marked *to be analyzed* in accordance with the aforementioned WRS procedure. The state of all other blocks remains unchanged; therefore, if other blocks were already marked to be analyzed, they will be analyzed in addition to the random blocks we have just set.
3. Blocks marked as *to be analyzed* are so using KDE; then, pixels classified as foreground are counted and the block set to *positive*/*negative* if the proportion of foreground pixels in the block is greater than a threshold value. If the threshold is high, fewer blocks will be eventually analyzed but the results on the edges of the objects have a higher chance to be wrong; if the threshold is low, the number of blocks that will be analyzed is higher but the quality of the results is improved. Therefore, we have used a very conservative 10% threshold in our tests.
4. If there are blocks marked as *positive* that have neighbours marked as *not analyzed*, these neighbours (and optionally more blocks within a certain radius) are changed *to be analyzed* and we go back again to step 3. Otherwise, the current frame is done. All remaining blocks marked as *not analyzed* are classified as background by default.
5. All blocks are set to *not analyzed* except those dictated by the temporal coherency criterion: blocks containing a ratio of predicted foreground over the threshold will be set *to be analyzed* in the next frame; we go back to step 2 for the next frame.

7. Results

The proposed strategy has been tested on a very large variety of indoor and outdoor sequences containing critical challenges for moving object detection.

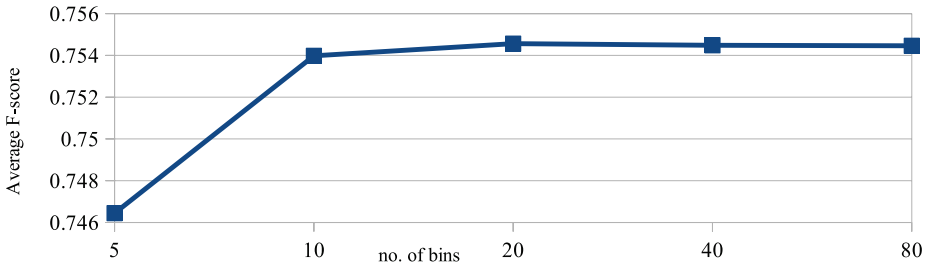

Figure 10: Average F-score across the LASIESTA database using only the background model for different numbers of bins for dynamic width switching.

These sequences have been extracted from three databases:

- SABS[2]: It is a very popular database that was proposed in [8] and has been recently used by many authors to test their algorithms ([9, 40, 22]). It is composed by synthetic sequences to evaluate seven different challenge situations. Every image in each sequence has a ground-truth mask associated. To emulate the noise introduced by real camera sensors, the sequences have had Gaussian noise added to the pixels values. Moreover, they have been created using a raytracing technology with global illumination to try and simulate scenes with realistic lighting.

- STAR[3]: It is composed by nine real video sequences recorded in different indoor and outdoor environments, published in [27]. In contrast to the SABS database, the ground-truth for each sequence consists in a subset of 20 images randomly selected. However, the challenges in this database have made it one of the most used by authors to test their moving object detection approaches ([15, 29, 22]).

- LASIESTA[4]: It is a new database, published in [14], that stands out among other databases due to the great amount of challenges it contains and because it is the only existing database with real videos that are fully annotated at both pixel-level and object-level.

We have compared our results to the best reported in the literature for each of the selected databases. The quality of the detections for the SABS and LASIESTA databases has been measured using the harmonic mean of the recall ($r = \frac{cd}{cd+md}$) and precision ($p = \frac{cd}{cd+fd}$), usually called F-score ($F = 2\frac{r \cdot p}{r+p}$), where cd is the number of correct detections, fd is the amount of false detections and md is the number of misdetections; for the STAR database we have used the average similarity score ($sim = \frac{cd}{cd+fd+md}$), which is the measure reported in the literature for this database.

[2] www.vis.uni-stuttgart.de/index.php?id=sabs
[3] perception.i2r.a-star.edu.sg/bk_model/bk_index.html
[4] www.gti.ssr.upm.es/data/LASIESTA

Table 1: Recall, precision and F-score values (as percentages) obtained in three key stages of the proposed strategy.

Database	BG			BG + FG			BG + FG + RoI		
	Recall	Precision	F	Recall	Precision	F	Recall	Precision	F
SABS	72.89	69.21	67.61	85.85	64.70	72.43	85.63	65.50	72.84
STAR	40.78	82.31	53.87	82.58	74.80	77.38	81.96	75.25	77.35
LASIESTA	79.22	75.30	75.48	91.95	69.80	78.15	91.81	73.46	80.51
Average	64.30	75.61	65.65	86.79	69.77	75.99	86.47	71.40	76.90

Figure 11: (a) Original images. Results with: (b) only background model; (c) adding foreground model; (d) adding the RoI analysis.

Table 2: Frames per second in three key stages of the proposed strategy.

Database	BG	BG + FG	BG + FG + RoI
SABS	10	9	25
STAR	30	18	28
LASIESTA	30	21	42

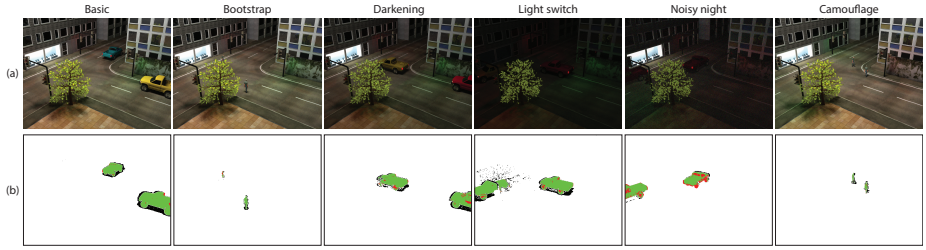

Figure 12: Some representative results obtained with the proposed detection strategy on the SABS database. (a) Original images. (b) Obtained detections.

All the experiments have been carried out on an NVIDIA GTX 580 GPU with 1.5 GiB RAM, coupled with an Intel Core i7-2600 with 16 GiB RAM.

In section 7.1, the adequate selection of the parameters described along the document is discussed. Then, in section 7.2 we analyze the performance of the proposed strategy in different cases. Finally, in section 7.3 we compare the results provided by our strategy with those obtained with other state-of-the-art alternatives. In order to enable other researchers to reproduce our reported results or compare our proposal with their own, we have made the results and source code available to the public[5].

7.1. Parameter selection

To reduce the influence of shadows and reflected light in the detections, both foreground and background models are obtained using the appearance vector described in [12], which is composed by the chromaticity (Rn, Gn) and the module of the gradient of the brightness, $|\nabla s|$.

In the case of background modelling it is only necessary to determine the number of the reference images and the spatial width assigned to the Gaussian kernels. The first one must be large enough to model the cyclical background changes along the sequences. We have set it as $T_\beta = 200$ for the sequences in both the STAR and LASIESTA databases; and as $T_\beta = 650$ for the sequences in the SABS database, since these sequences include much larger background cycles. Regarding the spatial width of the kernels, the performed experiments have shown that most false detections are avoided using a neighbourhood of 8-connected pixels ($\sigma^2_{\beta,R} = \sigma^2_{\beta,C} = \left(\sqrt{2}/3\right)^2$). Since larger spatial widths significantly increase the computational cost [11], this neighbourhood has been used for all the analyzed databases. The number of bins used for the appearance width estimation could be specified as discussed in section 4.1.1 but, as Fig. 10 shows, a fine estimation is not necessary: even very coarse partitions yield reasonable results and there is no reason to use more than twenty bins. Consequently, we have used this value across all experiments.

[5] www.gti.ssr.upm.es/data/

Table 3: *F*-score values (as percentages) obtained in the SABS database. Algorithm ranks are given by the numbers in brackets. The last column contains the average for all the tests.

Method	Basic	Dynamic background	Bootstrap	Darkening	Light switch	Noisy night	Camouflage	No camouflage	Average
1995-McFarlane [32]	61.4 (11)	48.2 (11)	54.1 (09)	49.6 (09)	21.1 (11)	20.3 (09)	73.8 (10)	78.5 (09)	50.9 (11)
1999-Stauffer [43]	80.0 (02)	70.4 (06)	64.2 (06)	40.4 (11)	21.7 (09)	19.4 (10)	80.2 (05)	82.6 (03)	57.4 (09)
2000-Oliver [35]	63.5 (10)	55.2 (10)	-	30.0 (13)	19.8 (12)	21.3 (08)	80.2 (04)	82.4 (04)	50.3 (12)
2000-McKenna [33]	52.2 (13)	41.5 (12)	30.1 (12)	48.4 (10)	30.6 (06)	09.8 (11)	62.4 (12)	65.6 (12)	42.6 (13)
2003-Li 1 [26]	76.6 (06)	64.1 (08)	67.8 (04)	70.4 (03)	31.6 (05)	04.7 (12)	76.8 (08)	80.3 (06)	59.0 (07)
2004-Kim [25]	58.2 (12)	34.1 (13)	31.8 (11)	34.2 (12)	-	-	77.6 (07)	80.1 (07)	52.7 (10)
2006-Zivkovik [48]	76.8 (04)	70.4 (05)	63.2 (07)	62.0 (07)	30.0 (07)	32.1 (06)	82.0 (02)	82.9 (02)	62.4 (04)
2008-Maddalena 1 [29]	76.6 (05)	71.5 (03)	49.5 (10)	66.3 (05)	21.3 (10)	59.6 (02)	79.3 (03)	81.1 (05)	59.0 (06)
2009-Barnich [2]	76.1 (07)	71.1 (04)	65.8 (05)	67.8 (04)	26.8 (08)	27.1 (07)	74.1 (09)	79.9 (08)	61.4 (05)
2013-Shimada [40]	72.3 (09)	62.3 (09)	70.8 (02)	57.7 (08)	33.5 (04)	47.5 (03)	-	-	57.4 (08)
2013-Cuevas [12]	75.9 (08)	72.9 (02)	55.7 (08)	65.3 (06)	57.2 (02)	33.6 (05)	73.6 (11)	74.4 (11)	63.6 (03)
2014-Haines [22]	**83.6 (01)**	**82.7 (01)**	**71.7 (01)**	73.6 (02)	49.9 (03)	34.6 (04)	**84.8 (01)**	**85.1 (01)**	70.8 (02)
Proposed	77.9 (03)	69.4 (07)	68.0 (03)	**77.8 (01)**	**63.7 (01)**	**70.5 (01)**	78.0 (06)	77.4 (10)	**72.8 (01)**

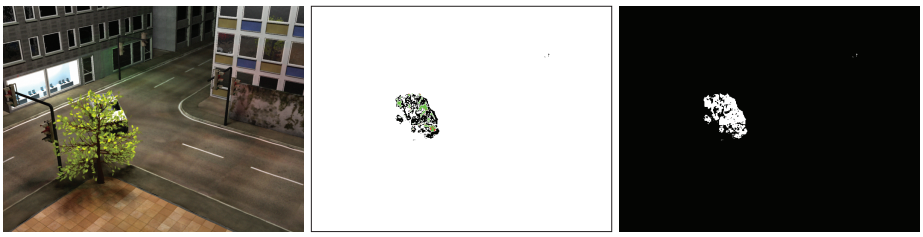

Figure 13: (a) Image from the SABS database where a moving object is partially occluded by a tree. (b) Detection obtained with the proposed strategy (similar colour notation to that used for previous figures). (c) Binary mask resulting from the detection.

Table 4: Similarity values (as percentages) obtained in the STAR database. Algorithm ranks are given by the numbers in brackets. The last column contains the average for all the tests.

Method	Bootstrap	Curtain	Fountain	Shopping Mall	Switch light	Trees	Water surface	Average
1999-Stauffer [43]	38.38 (5)	75.80 (4)	68.54 (3)	53.63 (5)	65.19 (3)	07.57 (7)	79.48 (4)	55.51 (5)
2004-Li 2 [27]	30.79 (6)	18.41 (7)	09.99 (7)	52.09 (6)	15.54 (7)	15.96 (6)	06.67 (7)	21.35 (7)
2007-Culibrk [15]	47.79 (4)	73.68 (5)	46.36 (5)	56.96 (4)	62.76 (4)	52.56 (4)	75.40 (5)	59.36 (4)
2008-Maddalena 1 [29]	60.19 (3)	81.78 (3)	65.54 (4)	**66.77 (1)**	64.89 (2)	69.60 (3)	82.47 (3)	70.18 (3)
2013-Cuevas [12]	17.09 (7)	23.17 (6)	41.50 (6)	25.89 (7)	28.86 (6)	37.79 (5)	39.21 (6)	30.50 (6)
2014-Haines [22]	60.24 (2)	82.03 (2)	**70.49 (1)**	65.22 (2)	57.94 (5)	**75.67 (1)**	**90.90 (1)**	71.78 (2)
Proposed	**64.10 (1)**	**84.12 (1)**	70.06 (2)	64.15 (3)	**66.28 (1)**	70.37 (2)	90.11 (2)	**72.74 (1)**

In the case of the foreground model, it is also necessary to set the number of reference images and the spatial width of the kernels. We have set the former as $T_\phi = 10$, which is enough in any sequence, since the foreground does not typically exhibit cyclical changes as the background may do. The spatial width of the kernels has been set slightly higher than that of the background to allow for some imprecision in the motion estimation, $\sigma^2_{\phi,R} = \sigma^2_{\phi,C} = (2/3)^2$. Finally, for the sequences in LASIESTA and STAR we have set a fixed appearance width of 0.02, which is sufficiently larger than the typical values for the background noise, which are in the order of 10^{-3} in the set of appearance components we have used. However, for the sequences in SABS this width has been slightly increased to 0.10, since it contains many illumination changes that induce persistent false detections if set too low.

7.2. Performance of the proposed strategy

To validate each of the stages of the proposed strategy, we have analyzed the results and their associated computational costs in three cases:

- BG: Modelling only the background.
- BG + FG: Modelling both background and foreground.
- BG + FG + RoI: Modelling both background and foreground and using the RoI module.

Table 1 summarizes the quality measures obtained in the three cases and some representative results are illustrated in Fig. 11. It can be observed that by adding the foreground modelling, the amount of misdetections is drastically reduced: the detections are more compact (fewer red pixels in Fig. 11.b) and the average recall increases significantly. Adding the RoI module (Fig. 11.c) eliminates small false detections (black pixels) whereas the amount of correct detections is maintained, resulting in a slight increase of the precision and similar recall. We have reported the raw results of our proposal without applying any post-processing and/or regularisation scheme to better illustrate the strengths of the proposed modelling strategy by itself.

Figure 14: Some representative results obtained with the proposed detection strategy on the STAR database. (a) Original images. (b) Obtained detections.

Table 2 shows average speed (frames per second) achieved in the three aforementioned configurations in the three evaluated databases. Note that, in order to be fair, all measurements are referred to the fractions of the sequences where all the models have been fully initialized and the system has reached a stationary regime, starting from image $I^{T_\beta+1}$, because previous images are much faster to evaluate simply because they have fewer data to process. Evaluation times of the background model depend on the number of used reference images, T_β, as expected (it is approximately constant because the amount of data and the number of operations is the same irrespectively of the contents of the images). The cost of the foreground model, on the other hand, depends on the fraction of foreground present in the scene. Finally, by adding the RoI module the cost of the background model is drastically reduced, allowing real-time operation in all the evaluated databases.

7.3. Comparison with other detection strategies

7.3.1. SABS

Some representative results obtained with the proposed strategy on the sequences of SABS are illustrated in Fig. 12, and the achieved F-score values and those reported by several alternative approaches are shown in Table 3. The published scores in [8] were obtained without postprocessing steps where applicable and adjusting only one parameter between sequences in each algorithm to maximize performance. However, to prove the high usability and adaptability of our proposal, we have chosen not to tune any parameter in a per-sequence basis and have used instead a single set of parameters across all sequences.

The proposed nonparametric strategy shows consistent performance in all sequences and obtains the best average F-score, even if it does not provide the best results in every category. Additionally, it is remarkable that in very dark scenarios (e.g. Darkening, Light switch and Noisy night) it is significantly better than previous approaches, since it is able to detect objects that are barely distinguishable from the background to the naked eye.

In some sequences (e.g. Dynamic Background) our F-score might be considered too low. However, as Fig. 13 shows, this is just because our spatiotemporal modelling strategy partially "fills" voids due to the occlusion of the tree leaves and results in a subjectively correct detection, even if it is objectively wrong according to the per-pixel provided ground-truth.

Table 5: *F*-score values (as percentages) obtained in the LASIESTA database. Algorithm ranks are given by the numbers in brackets. The last column contains the average for all the tests

Method	I_SI_01	I_SI_02	I_CA_01	I_CA_02	I_OC_01	I_OC_02	I_IL_01
1999-Stauffer [43]	84.09 (6)	82.47 (5)	88.11 (5)	77.33 (4)	95.00 (4)	82.71 (7)	34.98 (5)
2006-Zivkovic [48]	91.18 (4)	89.90 (2)	90.94 (3)	75.46 (5)	**98.80 (1)**	91.35 (3)	16.48 (7)
2008-Maddalena 1 [29]	89.28 (5)	84.65 (3)	**95.32 (1)**	73.94 (6)	98.03 (2)	84.66 (6)	85.33 (2)
2012-Maddalena 2 [30]	95.59 (2)	**94.09 (1)**	84.16 (7)	**87.31 (1)**	95.73 (3)	95.08 (2)	18.98 (6)
2013-Cuevas [12]	81.43 (7)	75.76 (7)	84.24 (6)	62.96 (7)	82.74 (6)	87.81 (5)	79.66 (3)
2014-Haines [22]	**96.22 (1)**	81.30 (6)	92.20 (2)	86.56 (2)	89.20 (5)	**95.26 (1)**	**88.61 (1)**
Proposed	92.08 (3)	84.03 (4)	90.62 (4)	78.26 (3)	70.13 (7)	86.00 (4)	64.52 (4)

Method	I_IL_02	I_MB_01	I_MB_02	I_BS_01	I_BS_02	O_CL_01	O_CL_02
1999-Stauffer [43]	23.92 (6)	83.42 (6)	69.40 (5)	35.55 (7)	36.94 (7)	89.22 (6)	84.57 (6)
2006-Zivkovic [48]	31.35 (5)	93.21 (4)	80.15 (3)	54.72 (3)	51.95 (4)	93.03 (3)	82.26 (7)
2008-Maddalena 1 [29]	37.50 (4)	84.73 (5)	67.61 (7)	40.23 (5)	44.65 (5)	89.85 (5)	85.47 (5)
2012-Maddalena 2 [30]	23.12 (7)	97.28 (2)	85.17 (2)	40.15 (6)	40.21 (6)	**96.57 (1)**	**97.60 (1)**
2013-Cuevas [12]	78.64 (2)	77.79 (7)	67.97 (6)	50.65 (4)	66.07 (2)	92.80 (4)	89.95 (4)
2014-Haines [22]	**81.22 (1)**	**98.16 (1)**	70.64 (4)	62.85 (2)	**73.33 (1)**	69.46 (7)	95.88 (2)
Proposed	65.23 (3)	95.43 (3)	**92.04 (1)**	**71.32 (1)**	61.56 (3)	95.08 (2)	90.45 (3)

Method	O_RA_01	O_RA_02	O_SN_01	O_SN_02	O_SU_01	O_SU_02	Average
1999-Stauffer [43]	74.35 (7)	82.75 (7)	73.69 (4)	47.24 (3)	61.77 (6)	83.04 (5)	69.53 (7)
2006-Zivkovic [48]	**85.86 (1)**	89.80 (3)	52.06 (6)	24.02 (5)	54.26 (7)	87.75 (3)	71.73 (6)
2008-Maddalena 1 [29]	82.52 (4)	85.88 (6)	69.77 (5)	45.95 (4)	74.67 (3)	85.62 (4)	75.28 (4)
2012-Maddalena 2 [30]	83.53 (3)	**95.91 (1)**	90.93 (2)	**71.16 (1)**	**87.42 (1)**	88.43 (2)	78.42 (2)
2013-Cuevas [12]	74.62 (6)	86.99 (5)	82.14 (3)	08.95 (6)	65.27 (5)	80.74 (6)	73.86 (5)
2014-Haines [22]	82.25 (5)	95.90 (2)	30.54 (7)	04.26 (7)	81.15 (2)	**90.21 (1)**	78.26 (3)
Proposed	84.53 (2)	88.86 (4)	**93.17 (1)**	62.56 (2)	67.74 (4)	76.69 (7)	**80.51 (1)**

7.3.2. STAR

The similarity values obtained for this database are shown in Table 4 and some representative results are illustrated in Fig. 14. We have discarded two of the nine test sequencies of this database that were encoded with extremely poor quality, since the proposed background modelling has not been designed to model pixel variations due to coding artifacts. Unlike SABS, this database allows using post-processing stages. Additionally, it allows tuning the algorithms to improve the quality of the results on each sequence. Again, our results have been obtained with a single set of parameters to prove the high usability and adaptability of our proposal.

The results in Fig. 14 show that our strategy is able to provide successful detections in all the challenges proposed in STAR: complex dynamic backgrounds, illuminations, changes, moving objects remaining static, etc. Additionally, as it can be seen in Table 4, it achieves the best average similarity.

Figure 15: Some representative results obtained with the proposed detection strategy on the LASIESTA database. (a) Original images. (b) Obtained detections.

7.3.3. LASIESTA

This database is composed by many more sequences than any of the two previous databases, and all of them are fully annotated. We have only selected those sequences without camera motion (a total of twenty sequences). Unlike in previous databases, the evaluation of the sequences in LASIESTA should be carried out without training period. All algorithms, including ours, have been run with a single set of parameters throughout the sequences.

The F-score values obtained for this database are shown in Table 5, and some representative results are illustrated in Fig. 15. Unlike with the other two databases, where two or three of the evaluated methods achieve the best results in each sequence, in LASIESTA there are up to five winning strategies (see the results highlighted in the table), due to the wide variety of challenges contained. Again, even if our strategy does not take the top spot in many sequences, it shows reasonable and consistent performance, obtaining the best overall F-score. Our weakest results (I_OC_01 or O_SU_02) come in sequences containing reflections or hard shadows cast by the moving objects since, unlike other researchers (e.g., [30, 48]) we do not apply any processing stage focused on shadow removal.

8. Conclusions

We have presented a high-quality, real-time nonparametric moving object detection strategy implemented in a GPU. The proposed strategy features robust spatio-temporal models of both the background and the foreground, the latter augmented with a novel tracking system based on a particle filter capable of dealing with a variable and unknown number of moving regions. The filter updates the positions of reference data to improve the relevance of reference samples of the foreground model and significantly cut down processing time; in addition, it also provides prior probability estimations for a Bayesian classifier that is able to combine models with different spatial widths and dissimilar spatial distributions, significantly improving detections. We have also presented a selective analysis strategy that automatically selects regions of interest in the input images, yielding equivalent results at a fraction of the computational cost.

We have validated our proposal by extensive testing on a large variety of sequences from three databases containing many challenges for motion detection and compared against many state-of-the-art methods. Although other methods may be able to score better in specific sequences, our results show good performance across a wide variety of scenarios without per-sequence tuning. This demonstrates the excellent usability and adaptability of our proposal, that also provides real-time performance thanks to massive parallelisation on a consumer-grade GPU.

Acknowledgments

This work has been partially supported by the Ministerio de Economía, Industria y Competitividad (AEI/FEDER) of the Spanish Government under projects TEC2013-48453 (MR-UHDTV) and TEC2016-75981 (IVME).

References

[1] Arulampalam, M. S., Maskell, S., Gordon, N., Clapp, T., 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Signal Process. 50 (2), 174–188.

[2] Barnich, O., Van Droogenbroeck, M., 2009. ViBe: a powerful random technique to estimate the background in video sequences. In: IEEE Int. Conf. Acoust., Speech and Signal Process. pp. 945–948.

[3] Berjón, D., Cuevas, C., Morán, F., García, N., 2013. GPU-based implementation of an optimized nonparametric background modeling for real-time moving object detection. IEEE Trans. Consum. Electron. 59 (2), 361–369.

[4] Bloisi, D. D., Pennisi, A., Iocchi, L., 2016. Parallel multi-modal background modeling. Pattern Recognition Letters, –.

[5] Bors, A. G., Nasios, N., 2009. Kernel bandwidth estimation for nonparametric modeling. IEEE Trans. Syst. Man Cybern. B 39 (6), 1543–1555.

[6] Bouwmans, T., 2014. Traditional and recent approaches in background modeling for foreground detection: An overview. Comput. Sci. Review 11, 31–66.

[7] Bouwmans, T., El Baf, F., Vachon, B., 2008. Background modeling using mixture of gaussians for foreground detection-a survey. Recent Patents Comput. Science 1 (3), 219–237.

[8] Brutzer, S., Höferlin, B., Heidemann, G., 2011. Evaluation of background subtraction techniques for video surveillance. In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition. pp. 1937–1944.

[9] Chang, H. J., Jeong, H., Choi, J. Y., 2012. Active attentional sampling for speed-up of background subtraction. In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition. pp. 2088–2095.

[10] Criminisi, A., Cross, G., Blake, A., Kolmogorov, V., 2006. Bilayer segmentation of live video. In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition. Vol. 1. pp. 53–60.

[11] Cuevas, C., Berjón, D., Morán, F., García, N., 2012. Moving object detection for real-time augmented reality applications in a GPGPU. IEEE Trans. Consum. Electron. 58 (1), 117–125.

[12] Cuevas, C., García, N., 2013. Improved background modeling for real-time spatio-temporal non-parametric moving object detection strategies. Image and Vision Computing 31 (9), 616–630.

[13] Cuevas, C., Martínez, R., García, N., 2016. Detection of stationary foreground objects: A survey. Computer Vision and Image Understanding 152, 41 – 57.

[14] Cuevas, C., Yáñez, E. M., García, N., 2016. Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA. Computer Vision and Image Understanding 152, 103 – 117.

[15] Culibrk, D., Marques, O., Socek, D., Kalva, H., Furht, B., 2007. Neural network approach to background modeling for video object segmentation. IEEE Trans. Neural Netw. 18 (6), 1614–1627.

[16] Douc, R., Cappé, O., 2005. Comparison of resampling schemes for particle filtering. In: Proc. Int. Symp. Image and Signal Process. and Anal. pp. 64–69.

[17] Elgammal, A., Duraiswami, R., Davis, L. S., 2001. Efficient non-parametric adaptive color modeling using fast Gauss transform. In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition. Vol. 2. pp. II–563.

[18] Elgammal, A., Duraiswami, R., Harwood, D., Davis, L. S., 2002. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proc. IEEE 90 (7), 1151–1163.

[19] Gallego, G., Cuevas, C., Mohedano, R., García, N., 2013. On the Mahalanobis distance classification criterion for multidimensional normal distributions. IEEE Trans. Signal Process. 61 (17), 4387–4396.

[20] Ge, W., Guo, Z., Dong, Y., Chen, Y., 2016. Dynamic background estimation and complementary learning for pixel-wise foreground/background segmentation. Pattern Recognition 59, 112 – 125, compositional Models and Structured Learning for Visual Recognition.

[21] Gong, M., Qian, Y., Cheng, L., 2015. Integrated foreground segmentation and boundary matting for live videos. IEEE Trans. Image Process. 24 (4), 1356–1370.

[22] Haines, T. S., Xiang, T., 2014. Background subtraction with Dirichlet process mixture models. IEEE Trans. Pattern Anal. Mach. Intell. 36 (4), 670–683.

[23] Hu, S., Poskitt, D. S., Zhang, X., 2012. Bayesian adaptive bandwidth kernel density estimation of irregular multivariate distributions. Computational Stat. & Data Anal. 56 (3), 732–740.

[24] Isard, M., Blake, A., 1998. Condensation—conditional density propagation for visual tracking. Int. J. Comput. Vision 29 (1), 5–28.

[25] Kim, K., Chalidabhongse, T. H., Harwood, D., Davis, L., 2004. Background modeling and subtraction by codebook construction. In: IEEE Int. Conf. Image Process. Vol. 5. pp. 3061–3064.

[26] Li, L., Huang, W., Gu, I. Y., Tian, Q., 2003. Foreground object detection from videos containing complex background. In: Proc. ACM Int. Conf. Multimedia. pp. 2–10.

[27] Li, L., Huang, W., Gu, I. Y.-H., Tian, Q., 2004. Statistical modeling of complex backgrounds for foreground object detection. IEEE Trans. Image Process. 13 (11), 1459–1472.

[28] Liang, D., Kaneko, S., Hashimoto, M., Iwata, K., Zhao, X., 2015. Co-occurrence probability-based pixel pairs background model for robust object detection in dynamic scenes. Pattern Recognition 48 (4), 1374 – 1390.

[29] Maddalena, L., Petrosino, A., 2008. A self-organizing approach to background subtraction for visual surveillance applications. IEEE Trans. Image Process. 17 (7), 1168–1177.

[30] Maddalena, L., Petrosino, A., 2012. The SOBS algorithm: what are the limits? In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition Workshop. pp. 21–26.

[31] Martel-Brisson, N., Zaccarin, A., 2008. Unsupervised approach for building non-parametric background and foreground models of scenes with significant foreground activity. In: ACM Workshop on Vision Networks for Behavior Analysis. VNBA '08. ACM, New York, NY, USA, pp. 93–100.

[32] McFarlane, N. J., Schofield, C. P., 1995. Segmentation and tracking of piglets in images. Machine Vision and Applications 8 (3), 187–193.

[33] McKenna, S. J., Jabri, S., Duric, Z., Rosenfeld, A., Wechsler, H., 2000. Tracking groups of people. Comput. Vision and Image Understanding 80 (1), 42–56.

[34] Mihaylova, L., Carmi, A. Y., Septier, F., Gning, A., Pang, S. K., Godsill, S., 2014. Overview of bayesian sequential monte carlo methods for group and extended object tracking. Digital Signal Process. 25, 1–16.

[35] Oliver, N. M., Rosario, B., Pentland, A. P., 2000. A Bayesian computer vision system for modeling human interactions. IEEE Trans. Pattern Anal. Mach. Intell. 22 (8), 831–843.

[36] Palmero, C., Clapés, A., Bahnsen, C., Møgelmose, A., Moeslund, T. B., Escalera, S., 2016. Multi-modal RGB–depth–thermal human body segmentation. International Journal of Computer Vision, 1–23.

[37] Paragios, N., Ramesh, V., 2001. A MRF-based approach for real-time subway monitoring. In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition. Vol. 1. pp. I–1034–I–1040.

[38] Piccardi, M., 2004. Background subtraction techniques: a review. In: IEEE Int. Conf. Syst., Man and Cybern. Vol. 4. pp. 3099–3104.

[39] Sheikh, Y., Shah, M., 2005. Bayesian modeling of dynamic scenes for object detection. IEEE Trans. Pattern Anal. Mach. Intell. 27 (11), 1778–1792.

[40] Shimada, A., Nagahara, H., Taniguchi, R.-i., 2013. Background modeling based on bidirectional analysis. In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition. pp. 1979–1986.

[41] Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M., 2014. Visual tracking: an experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. 36 (7), 1442–1468.

[42] Sobral, A., Vacavant, A., 2014. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. Comput. Vision and Image Understanding 122, 4–21.

[43] Stauffer, C., Grimson, W. E. L., 1999. Adaptive background mixture models for real-time tracking. In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition. Vol. 2. pp. 246–252.

[44] Stockman, G., Shapiro, L. G., 2001. Computer Vision, 1st Edition. Prentice Hall PTR.

[45] Varadarajan, S., Miller, P., Zhou, H., 2015. Region-based mixture of gaussians modelling for foreground detection in dynamic scenes. Pattern Recognition 48 (11), 3488 – 3503.

[46] Wixson, L., 2000. Detecting salient motion by accumulating directionally-consistent flow. IEEE Trans. Pattern Anal. Mach. Intell. 22 (8), 774–780.

[47] Zhang, T., Liu, S., Ahuja, N., Yang, M.-H., Ghanem, B., 2015. Robust visual tracking via consistent low-rank sparse learning. International Journal of Computer Vision 111 (2), 171–190.

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