Synergistic Change Detection and Tracking

Samuele Salti, *Member, IEEE*, Alessandro Lanza, and Luigi Di Stefano, *Member, IEEE*

***Abstract*— Visual tracking in image streams acquired by static cameras is usually based on change detection and recursive Bayesian estimation, such an approach laying at the core of many practical applications. Yet, the interaction between the change detector and the Bayesian filter is typically designed heuristically. Differently, this paper develops a sound framework to model and implement a bidirectional communication flow between the two processes. In our Bayesian loop, change detection provides well-defined observation likelihood to the recursive filter and the filter prediction provides an informative prior to the change detector, which deploys Bayesian reasoning alike. The loop is developed for the two major variants of Bayesian filters used in tracking, namely the Kalman filter and the particle filter. Experiments on publicly available videos and a novel challenging data set show that the proposed interaction scheme outperforms several state-of-the-art trackers.**

***Index Terms*— Image sequence analysis, Kalman filters, motion detection, particle filters real-time tracking, video surveillance, video tracking.**

I. Introduction

THIS paper investigates visual tracking with static cameras. The usual approach [1]–[11] in such scenario is to ground tracking on change detection, i.e., on a process that labels every pixel as changed (i.e., a target pixel) or unchanged (i.e., a background pixel) with respect to a static background. In these proposals, the interaction between the tracking module and the change detection module is key to success. Yet, sound modeling of the information flow between the change detector and the tracker did not receive adequate attention from researchers in the field, the interaction between the two modules relying typically on *ad hoc* algorithms. For example, a standard approach is to threshold the change detection output and then perform some sort of blob analysis to steer the tracker [2], [3]. As such, the overall computation tends to be sensitive to parameters tuned heuristically, such as change detection thresholds or acceptable aspect ratio of blobs, an issue that hinders both the generality of research results and deployment in real world applications.

Bayesian inference offers a sound framework to model and solve the tracking problem. Recursive Bayesian estimation (RBE) filters [12] are the standard tool to tackle the filtering and data association problem [13] in visual tracking. Formulating tracking as a probabilistic inference problem considers seamlessly the uncertainty on target state caused by typical nuisances, such as illumination and pose changes, sensor noise, complex motion patterns, deformable targets, clutter, distractors, and occlusions.

Bayesian reasoning can be deployed to solve the problem of change detection alike. In particular, change detection can be formalized as the computation of the posterior probability for each pixel to be changed or unchanged given the current frame and the reference background. Such a modeling opens up the change detection module to the possibility of plugging in seamlessly prior information on pixels being changed, either from external modules or from known statistics.

Accordingly, given a Bayesian change detector and a generic recursive Bayesian filter, we develop a principled framework whereby both algorithms can virtuously influence each other within a Bayesian loop. In particular, on one side of the loop, the output of the change detection is used without thresholding to provide the observation likelihood to the RBE filter via a reasoning based on an *a priori* model of the expected output of the change detection given a tracker state; on the other side, the RBE prediction for the current frame is used to provide a feedback to the Bayesian change detector by defining an informative prior, whose derivation exploits the same *a priori* model. The idea of letting the tracker provide a feedback to change detection is inspired by the emergence of cognitive feedback in computer vision [14], in particular, by the observation that information from higher level modules can usefully influence low-level vision.

In this paper, we present the practical instantiation of such ideas for the two prominent RBE filters, namely the Kalman filter and the particle filter (PF). Our contributions include: 1) model-based analysis of the change detector output, so as to define a measurement with uncertainty for the Kalman filter and a likelihood to reweight the particles for the PF, without resorting to any thresholding and blob analysis and 2) derivation of the cognitive feedback from the probability density function (PDF) of the state predicted by the RBE filter for the current frame to the Bayesian change detector. We demonstrate superior performance of the proposed algorithm with respect to the most recent and prominent literature proposals via extensive experiments on publicly available and novel challenging videos. A preliminary version of this framework was presented in [15].

This paper is organized as follows. After a brief review of relevant previous papers in Section II, we present the overall Bayesian loop for Kalman and particle filtering in Section III. Then, Section IV describes how to realize the cognitive feedback. Section V-A and V-B address the other side of the loop, i.e., the definition of the likelihood from the

Manuscript received April 16, 2014; revised July 24, 2014; accepted August 28, 2014. Date of publication September 8, 2014; date of current version April 2, 2015. This paper was recommended by Associate Editor J. Ostermann.

S. Salti and L. Di Stefano are with the Department of Computer Science and Engineering, University of Bologna, Bologna 40126, Italy (e-mail: samuele.salti@unibo.it; alessandro.lanza2@unibo.it).

A. Lanza is with the Department of Mathematics, University of Bologna, Bologna 40126, Italy (e-mail: luigi.distefano@unibo.it).

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

Digital Object Identifier 10.1109/TCSVT.2014.2355695

change detection output, in the Kalman case and the PF case, respectively. Section VI presents a thorough quantitative experimental evaluation of our proposal.

II. Related Work

The main difficulty with change detection consists in discerning changes of the scene in presence of spurious intensity variations yielded by nuisances, such as noise, gradual, or sudden illumination changes, dynamic adjustments of camera parameters (e.g., autoexposure and autogain). Many different algorithms for dealing with these issues have been proposed (see [16] for a recent survey).

A first class of popular algorithms based on statistical per-pixel background models, such as mixture of Gaussians [4] or kernel-based nonparametric models [17], are effective in case of noise and gradual illumination changes (e.g., due to the time of the day). Unfortunately, though, they cannot deal with those disturbs causing sudden intensity changes (e.g., a light switch), yielding in such cases many false positives.

A second class of algorithms relies on *a priori* modeling the possible spurious intensity changes over small image patches yielded by disturbs. Following this idea, a pixel from the current frame is classified as changed, if the intensity transformation between its local neighborhood and the corresponding neighborhood in the background cannot be explained by the chosen *a priori* model. As a result, gradual as well as sudden photometric distortions do not yield false positives if they are explained by the model. Thus, the main issue concerns the choice of the *a priori* model: the more restrictive such a model, the higher is the ability to detect changes (sensitivity), but the lower is robustness to disturbs (specificity). Some proposals assume disturbs to yield linear intensity transformations [18], [19]. Nevertheless, as discussed in [20], many nonlinearities may arise in the image formation process, so that a less constrained model is often required to achieve adequate robustness. Hence, other algorithms adopt order-preserving models, i.e., they assume monotonic nondecreasing intensity transformations [20]–[22].

Classical works on blob tracking based on change detection are $\mathcal{W}4$ [2] and the system developed at the video surveillance and monitoring (VSAM) group of CMU [3]. In these systems, the output of the change detector is thresholded and a connected component analysis is carried out to identify moving regions (blobs). A first- or second-order dynamical model of every tracked object is used to predict its position in the current frame from previous ones. Positions are then refined by matching predictions to the output of the change detection. In the work of the VSAM group [3], any blob whose centroid falls within a neighborhood of the target predicted position is considered for matching. Matching is performed as correlation of an appearance template of the target to the changed pixels, and the position corresponding to the best correlation is selected as the new position for the object. In $\mathcal{W}4$ [2], the new position is that corresponding to the maximum of the binary edge correlation between the current and previous silhouette edge profiles. In these systems, the interaction between tracking and change detection is limited, tracking is not formalized in the context of RBE, change detection depends on hard thresholds, and *ad hoc* rules incorporating many parameters are used to derive a new measurement from the change detection output or to update the object position, (e.g., a set of heuristics is used to deal with the issue of several blobs related to the same object).

References [4] and [6] are examples of blob trackers based on change detection, where the RBE framework is used in the form of the Kalman filter. Yet, the proposed approaches lack a truly probabilistic treatment of the change detection output. In practice, covariance matrices defining measurement and process uncertainties are constant, and the filter evolves toward its steady state regardless of the quality of the measurements obtained from change detection. *A posteriori* covariance matrices are sometimes deterministically increased by the algorithms, but this is mainly a shortcut to implement track management: if there is no match for the track in the current frame uncertainties are increased and if *a posteriori* uncertainties on state gets too high, the track is discarded.

BraMBLe [7] is one of the most famous attempts to integrate RBE in the form of a PF with a statistical treatment of background (and foreground) models. It proposes a multiblob likelihood function that, given the frame and the background model, allows the system to reason probabilistically on the number of people present in the scene as well as on their positions. The main limitations are the use of a calibrated camera with reference to the ground plane and the use of a foreground model learned offline. While the former can be reasonable, although cumbersome, the use of foreground models is always troublesome in practice, given the exceedingly high intra-class variability of target appearances. A recent approach based on the PF in static camera scenarios is the real time compressive sensing tracker [23], which relies on the sparse signal recovery property of the ℓ_1-norm minimization to define a robust observation likelihood as the residual of the reconstruction with foreground and background patches. None of these proposals provides cognitive feedback from the PF to influence change detection.

Sorts of cognitive feedbacks from tracking to change detection have been used, so far only to deal with background maintenance and adaptive background modeling. For example, Taycher *et al.* [24] propose a method based on approximate inference on a dynamic Bayesian network that simultaneously solves tracking and background model updating for every frame. Nevertheless, as discussed by the authors, this proposal does not take advantage of models of foreground motion as our algorithm does, although this would allow for better estimation of both the background model and the background/foreground labels, because doing so will severely complicate inference. Another example of background maintenance is [25], where positive and negative feedbacks from high-level modules (a stereo-based people detector and tracker, a detector of rapid changes in global illumination, camera gain, and camera position) are used to update the parameters of the Gaussian distributions in the Gaussian mixture model used as background. These feedbacks come in the form of pixel-wise positive or negative real number maps that are gen-

TABLE I
NOTATION

$\mathbf{f}_k$	video frame at time k
$\mathbf{b}_k$	background at time k
w, h	frame width and height, respectively
f_k^{ij}	intensity at pixel (i,j) of frame $\mathbf{f}_k$
$\mathbf{x}_k$	state of the RBE filter at time k
$\mathbf{z}_k$	measurement for the RBE filter at time k
$\mathbf{z}_{1:k} = (\mathbf{z}_1, \ldots, \mathbf{z}_k)$	measurements up to time k
$(\mathbf{x}_k^{(n)}, w_k^{(n)})$	particle with state $\mathbf{x}_k^{(n)}$ and weight $w_k^{(n)}$
(i_k^b, j_k^b)	barycenter of the target bounding box
w_k, h_k	dimensions of the target bounding box
i_k^L, j_k^T, i_k^R, j_k^B	min/max i and j coordinates defining the target bounding box
$B(\mathbf{x}_k)$	set of pixels within the bounding box defined by state $\mathbf{x}_k$
$\bar{B}(\mathbf{x}_k)$	set of pixels outside the bounding box defined by state $\mathbf{x}_k$
c_k^{ij}	r. v. for the event "pixel (i,j) at time k is changed wrt to the background"
$\mathbf{c}_k = \left[c_k^{ij}\right]$	change mask at time k
$\mathbf{p}_k = \left[p(c_k^{ij} = \mathcal{C} \mid b_k^{ij}, f_k^{ij})\right]$	change map at time k

erated as sum of the contributions of the high-level modules and are thresholded to establish whether a pixel should be used to update the background. Contributions from the high-level modules are heuristically determined.

A complementary approach to tracking deals with modeling the target rather than the background, e.g., by color or intensity histograms [13], [26], [27], or by local features [28], the latter being a popular choice when tracking camera motion for the purpose of 3-D reconstruction or mapping [29]. Early trackers kept the target appearance model fixed throughout the sequence, whereas focusing on robust localization and matching strategies. Conversely, several recent methods have been proposed that let the target model evolve over time, so as to adapt it to geometric and photometric appearance changes [30]–[35]. Among these, a family of successful methods relies on boosting to create a discriminative classifier that can be trained online to separate the target from the background [36]–[38]. Such recent trackers usually do not deploy any stochastic filter to enforce temporal consistency, i.e., they implicitly assume a constant position motion model and then select as the new state that yielding the maximum confidence in a neighborhood of the previous state. References [39]–[41] present recent surveys and evaluations of trackers modeling the foreground appearance.

III. Bayesian Loop Overview

First, we present the assumptions used to model RBE and Bayesian change detection separately, then we introduce the framework of the proposed Bayesian loop. The notation used throughout this paper is summarized in Table I.

A. RBE Model

The RBE [12] aims at hidden state estimation from noisy measurements in discrete-time systems. The solution is obtained recursively: given the PDF of the state at time $k-1$ conditioned on all previous measurements, $p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1})$, and the availability of a new measurement, $\mathbf{z}_k$, a new estimate for the PDF at time k is computed. As detailed in [12], a general but conceptual solution can be obtained in two steps: 1) prediction and 2) update. In the prediction stage, the Chapman–Kolmogorov equation is used to propagate the belief on the state from time $k-1$ to time k

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}. \quad (1)$$

This usually corresponds to spreading the belief on the state, due to the increasing distance in time from the last measurement. In the update stage, the PDF is sharpened again using the current measurement $\mathbf{z}_k$ and the Bayes' rule

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \propto p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}). \quad (2)$$

This conceptual solution is analytically solvable only in a few cases. A notable one is when the state dynamics and measurement equations are linear and uncertainties are Gaussian. In this situation, the optimal solution is given by the Kalman filter [42].

Within the RBE framework, PFs provide a tractable approximate solution for the general nonlinear/non-Gaussian case. The approximation comes from representing the PDF of the state vector $\mathbf{x}_k$ at time k as the sum of a finite set of weighted samples $(\mathbf{x}_k^{(n)}, w_k^{(n)})$ called particles

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \approx \sum_{n=1}^{N} w_k^{(n)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(n)}) \quad (3)$$

where $\delta(\cdot)$ is the Dirac delta function. In the case of particle filtering, the Chapman–Kolmogorov equation used in the prediction stage to propagate the belief on the state from time $k-1$ to time k is realized by sampling from the so-called proposal distribution to obtain the new particle states $\mathbf{x}_k^{(n)}$, $n = 1, \ldots, N$, thus producing the PDF of the predicted state

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) \approx \sum_{n=1}^{N} w_{k-1}^{(n)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(n)}). \quad (4)$$

The proposal distribution is used, because it is in general hard to sample from the posterior. We rely on the standard approach of using the transition model $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ as proposal distribution. In the update stage, the likelihood of the current observation $\mathbf{z}_k$ is used to compute the posterior (3) by updating the weights

$$w_k^{(n)} \propto w_{k-1}^{(n)} p(\mathbf{z}_k \mid \mathbf{x}_k^{(n)}). \quad (5)$$

Finally, if MAP estimation is performed, the particle with the greatest weight is chosen as the estimate of the current state $\mathbf{x}_k$. More details on RBE can be found in [12].

As in many recent proposals [38], [43], [44], we assume a rectangular model for the tracked object. Hence, the state of the RBE tracker, $\mathbf{x}_k$, comprises at least four variables

$$\mathbf{x}_k = \left\{ i_k^b, j_k^b, w_k, h_k, \ldots \right\} \quad (6)$$

where (i_k^b, j_k^b) are the coordinates of the barycenter of the rectangle, and w_k and h_k are its dimensions. Of course,

the state internally used by the tracker may beneficially include other kinematic variables (velocity, acceleration, and so on), but only position and size of the target influence the frame-by-frame interaction between change detection and the RBE filter. Hence, other variables are not used in the reminder of the presentation of the algorithm, though they can be used internally by the RBE filter, and are indeed used in our implementation (Section VI).

Whenever convenient, we will also represent the bounding box by its minimum and maximum coordinates according to the variables $i_k^L,\ j_k^T,\ i_k^R,\ j_k^B$

$$\begin{bmatrix} i_k^L \\ i_k^R \end{bmatrix} = \mathbf{A} \begin{bmatrix} i_k^b \\ w_k \end{bmatrix}, \quad \begin{bmatrix} j_k^T \\ j_k^B \end{bmatrix} = \mathbf{A} \begin{bmatrix} j_k^b \\ h_k \end{bmatrix} \tag{7}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & -\frac{1}{2} \\ 1 & \frac{1}{2} \end{bmatrix}. \tag{8}$$

B. Bayesian Change Detection

Inspired by [45], we propose a change detection approach that, instead of assuming *a priori* the model of intensity changes caused by disturbs, learns it online together with the model of intensity changes yielded by foreground objects. In particular, at each new frame, a binary Bayesian classifier is trained and then used to discriminate between pixels sensing a scene change due to foreground objects and pixels sensing a spurious intensity variation due to disturbs. Online learning of the models holds the potential for deploying on a frame-by-frame basis models as restrictive as needed to discriminate between the two classes, so that the algorithm can exhibit high sensitivity without a significant loss of specificity.

Every pixel (i, j) of the image is modeled as a categorical Bernoulli-distributed random variable c_k^{ij}, with the two possible realizations $c_k^{ij} = \mathcal{C}$ and $c_k^{ij} = \mathcal{U}$ indicating, respectively, the event of pixel (i, j) in frame k being changed or unchanged with respect to the background. In the following, we refer to the $w \times h$ matrices $\mathbf{c}_k = [\, c_k^{ij} \,]$ and $\mathbf{p}_k = [\, p\,(c_k^{ij} = \mathcal{C} \,|\, b_k^{ij}, f_k^{ij})\,]$ containing the random variables and the parameters uniquely defining their Bernoulli distribution as change mask and change map, respectively. The change mask and the change map take values, respectively, in the $(w \times h)$-dimensional spaces $\Theta = \{\mathcal{C}, \mathcal{U}\}^{w \times h}$ and $\mathbf{\Omega} = [0, 1]^{w \times h}$. Given a change map, a change mask is obtained by thresholding it with a chosen confidence value.

At each new frame, a binary Bayesian classifier is trained using as feature vector the pair of intensities (b_k^{ij}, f_k^{ij}) observed at a pixel in the background and frame, respectively. This allows for computing the change map by letting each pixel take the *a posteriori* value of the probability of being changed

$$\begin{aligned} p_k^{ij} \doteq p\big(c_k^{ij} = \mathcal{C} \,|\, b_k^{ij}, f_k^{ij}\big) &= \frac{p\big(c_k^{ij} = \mathcal{C}\big)\, p\big(b_k^{ij}, f_k^{ij} \,|\, c_k^{ij} = \mathcal{C}\big)}{p\big(b_k^{ij}, f_k^{ij}\big)} \\ &= \frac{1}{1 + \dfrac{p\left(c_k^{ij} = \mathcal{U}\right)\, p\left(b_k^{ij}, f_k^{ij} \,|\, c_k^{ij} = \mathcal{U}\right)}{p\left(c_k^{ij} = \mathcal{C}\right)\, p\left(b_k^{ij}, f_k^{ij} \,|\, c_k^{ij} = \mathcal{C}\right)}}. \end{aligned} \tag{9}$$

Either a noninformative prior can be used, i.e., $p(c_k^{ij} = \mathcal{C}) = p(c_k^{ij} = \mathcal{U}) = 0.5$, or this information may be provided by an external module. We propose to deploy the prediction of the RBE filter for the current frame to attain an informative prior, thus creating a Bayesian loop.

To train the classifier, we have to estimate the likelihoods $p(b_k^{ij}, f_k^{ij} \,|\, c_k^{ij} = \mathcal{C})$ and $p(b_k^{ij}, f_k^{ij} \,|\, c_k^{ij} = \mathcal{U})$ for the current frame. First, we carry out a preliminary classification of pixels by means of a very efficient neighborhood-based change detection algorithm. For a generic pixel (i, j), let the intensity differences between the mth pixel of the 3×3 neighborhood and the central pixel in the background and in the current frame be, respectively, $db_m^{ij} = b^m - b^{ij}$ and $df_m^{ij} = f^m - f^{ij}$ and let the pixel in the neighborhood yielding the maximum absolute value of the background intensity difference be

$$\bar{m}_{ij} = \underset{m=1,\ldots,8}{\operatorname{argmax}}\ \operatorname{abs}(db_m^{ij}) \tag{10}$$

A preliminary change mask $\tilde{\mathbf{c}}_k$ is computed by classifying each pixel as unchanged if the sign of the intensity differences $db_{\bar{m}_{ij}}^{ij}$ and $df_{\bar{m}_{ij}}^{ij}$ is the same, changed otherwise

$$\tilde{c}_k^{ij} = \begin{cases} \mathcal{U} & \text{if } db_{\bar{m}_{ij}}^{ij} \cdot df_{\bar{m}_{ij}}^{ij} \geq 0 \\ \mathcal{C} & \text{otherwise} \end{cases} \tag{11}$$

this algorithm is a simplified version of that proposed in [20] and exhibits $O(N)$ complexity. The computation of $\bar{m}_{ij}$ for each pixel by using (10) can be performed offline after background initialization or update. Furthermore, the algorithm is parameter-free.

Then, the preliminary change mask is used to label each pixel to create a training set out of the current frame. The two likelihood distributions are estimated from this training set as follows:

$$p\big(b_k^{ij}, f_k^{ij} \,|\, c_k^{ij} = \mathcal{C}\big) = \frac{h_{\mathcal{C}}\big(b_k^{ij}, f_k^{ij}\big)}{N_{\mathcal{C}}} \tag{12}$$

$$p\big(b_k^{ij}, f_k^{ij} \,|\, c_k^{ij} = \mathcal{U}\big) = \frac{h_{\mathcal{U}}\big(b_k^{ij}, f_k^{ij}\big)}{N_{\mathcal{U}}} \tag{13}$$

where $N_{\mathcal{C}}$ ($N_{\mathcal{U}}$) is the number of pixels labeled as changed (unchanged) in the preliminary change map, $h_{\mathcal{C}}(b_k^{ij}, f_k^{ij})$ ($h_{\mathcal{U}}(b_k^{ij}, f_k^{ij})$) is the 2-D joint histograms of background versus frame intensity computed by considering the pixels labeled as changed (unchanged) in the preliminary change mask. Both histograms $h_{\mathcal{C}}(b_k^{ij}, f_k^{ij})$ and $h_{\mathcal{U}}(b_k^{ij}, f_k^{ij})$ are smoothed by averaging through a moving window of fixed size. Smoothing allows for correcting errors introduced by wrong-labeled training data in the preliminary rough labeling as well as for enforcing a small amount of spatial consistency among labels, under the hypothesis that pixels close to each other show similar intensity values both in the foreground and in the background.

C. Bayesian Loop

The overall tracking loop is shown in Fig. 1, along with a classical change detection-based tracker. The idea is to define an *a priori* model for the ideal change mask $\mathbf{c}_k$ given that the

Algorithm 1 Bayesian Loop algorithm with the Kalman Filter

Input: a sequence of graylevel images $\{\mathbf{f}_k\}_{k=1}^{L}$, a background model $\mathbf{b}$, an initial state $\mathbf{x}_0$, parameters K_1 and K_2
Output: a trajectory in the state space $\{\mathbf{x}_k\}_{k=1}^{L}$

define $\bar{m}_{ij}$ from $\mathbf{b}$ according to (10)
for $k = 1$ **to** L **do**
  // RBE predict
  obtain the PDF of the predicted state for the current frame $p^-(\mathbf{x}_k)$ by using Kalman Filter predict rule
  // Cognitive feedback
  estimate an informative prior $[p(c_k^{ij} = \mathcal{C})]$ by using equation (17), where I_k^{ij} is computed with (20)
  // Change Detection
  estimate preliminary change map $[\tilde{c}_k^{ij}]$ from $\mathbf{f}_k$ according to (11)
  estimate likelihoods according to (12) and (13)
  smooth likelihoods with 5x5 average filter
  compute *a-posteriori* change map $[p_k^{ij}]$ according to (9) and using the informative prior from the cognitive feedback
  // Extract observations from the change map
  compute log $p(\mathbf{x}_k)$ from p_k^{ij} by using (27)
  compute $\widehat{\boldsymbol{\mu}}_k$ by using (30)
  compute $\widehat{\sigma}_k^2(\mathbf{x})$ by averaging (33) over a neighborhood of $\widehat{\boldsymbol{\mu}}_k$
  compute $\mathbf{z}_k$ and $\mathbf{R}_k$ according to (34)
  // RBE update
  update the state estimate $\mathbf{x}_k$ by using $\mathbf{z}_k$ and $\mathbf{R}_k$ in the Kalman Filter update rule
end for

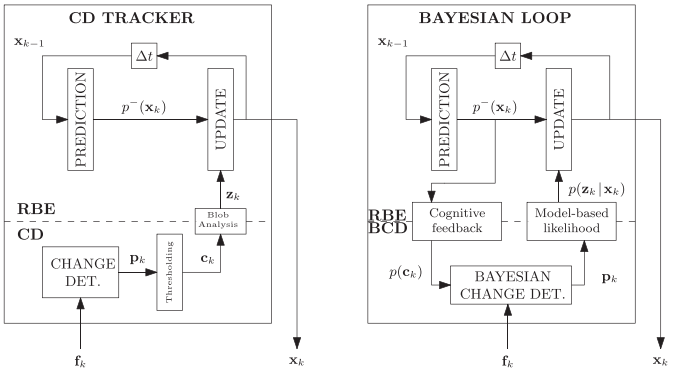

Fig. 1. Left: overview of a classical change detection-based tracker. Right: proposed Bayesian loop.

target bounding box $\mathbf{x}_k$ is known, so as to model soundly the interaction between the RBE filter and the change detection. Based on such a model, we can both define a principled and robust method to derive a likelihood for the RBE filter from the change detection output, as well as create an information flow from the tracker to the change detection, so to influence the latter in judging more likely to be changed pixels nearby the predicted target position.

To define such Bayesian loop, we need the model for the ideal change map $\mathbf{c}_k$ given that the target bounding box $\mathbf{x}_k$ is known. We assume that there is no spatial correlation between pixels (i.e., we assume that the random variables c_k^{ij} are conditionally independent given a bounding box $\mathbf{x}_k$), and that the probability that a pixel is changed inside and outside of the bounding box is constant (Fig. 2). Formally

$$p(\mathbf{c}_k \mid \mathbf{x}_k) = \prod_{ij} p(c_k^{ij} \mid \mathbf{x}_k) \tag{14}$$

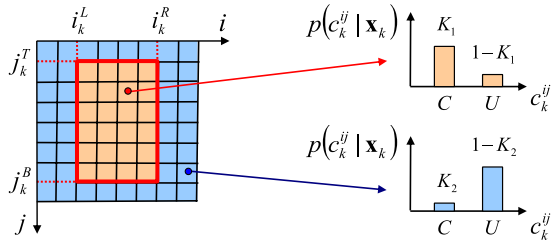

Fig. 2. Model for the change map given the target bounding box $\mathbf{x}_k = (i_k^L, j_k^T, i_k^R, j_k^B)$.

$$p\left(c_k^{ij} = \mathcal{C} \mid \mathbf{x}_k\right) = \begin{cases} K_1 & \text{if } (i,j) \in B(\mathbf{x}_k) \\ K_2 & \text{otherwise} \end{cases} \tag{15}$$

where $1 \geq K_1 \geq K_2 \geq 0$. Based on such a model, we proceed to derive the two sides of the loop in the following sections. For the sake of clarity, the main steps of the proposed algorithm are highlighted in Algorithms 1 and 2 for the case of the Kalman and the PF, respectively.

IV. Cognitive Feedback

The aim of this side of the loop is to compute an informative prior for the Bayesian change detector given the state predicted by the RBE filter. Formally, given the PDF of the predicted state $p^-(\mathbf{x}_k) \doteq p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})$ computed by the RBE filter, the output of this part of the algorithm is the prior probability $p(c_k^{ij} = \mathcal{C})$ for each pixel (i, j) of the current frame to be changed with respect to the background. A visual example of the cognitive feedback module is shown in Fig. 3, where brighter intensities denote higher priors.

In principle, all the information that can flow from the RBE filter to the Bayesian change detector and vice versa is subsumed by the joint probability distribution $p(\mathbf{x}_k, \mathbf{c}_k)$ of the target bounding-box and the change mask. Hence, the

Algorithm 2 Bayesian Loop algorithm with the PF

Input: a sequence of graylevel images $\{\mathbf{f}_k\}_{k=1}^{L}$, a background model $\mathbf{b}$, an initial state $\mathbf{x}_0$, parameters K_1 and K_2, number of particles N

Output: a trajectory in the state space $\{\mathbf{x}_k\}_{k=1}^{L}$

define $\bar{m}_{ij}$ from $\mathbf{b}$ according to (10)
for $k = 1$ **to** L **do**
  // RBE predict
  obtain the PDF of the predicted state for the current frame $p^-(\mathbf{x}_k)$ by sampling from the proposal distribution
  // Cognitive feedback
  estimate an informative prior $p(c_k^{ij} = \mathcal{C})$ by using equation (17) where I_k^{ij} is computed with (21)
  // Change Detection
  estimate preliminary change map $[\tilde{c}_k^{ij}]$ from $\mathbf{f}_k$ according to (11)
  estimate likelihoods according to (12) and (13)
  smooth likelihoods with 5x5 average filter
  compute *a-posteriori* change map $[p_k^{ij}]$ according to (9) and using the informative prior from the cognitive feedback
  // Extract observations from the change map
  Compute integral images of $\sqrt{p_k^{ij}}$ and $\sqrt{1 - p_k^{ij}}$
  for $n = 1$ **to** N **do**
    compute $p(\mathbf{z}_k \mid \mathbf{x}_k^{(n)})$ by using (38) and the integral images
  end for
  // RBE update
  update the weights $w_k^{(n)}$ by using (5)
  update the MAP state estimate $\mathbf{x}_k \leftarrow \mathbf{x}_k^{\bar{n}}$, where $\bar{n} = \underset{n=1,\ldots,N}{\operatorname{argmax}}\, w_k^{(n)}$
  perform particle resampling
end for

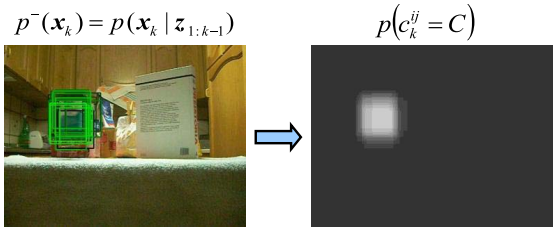

Fig. 3. Input (left) and output (right) of the cognitive feedback module in the case of particle filtering. Left: brighter the bounding boxes, the greater the weight of the corresponding particles. Right: brighter the pixels, the greater the predicted probability to belong to the tracked object. Though here the number of particles is limited to 20 for the sake of visualization, a much higher number is used in our implementation. This leads to smoother estimates of the map $p(c_k^{ij} = \mathcal{C})$.

information flow from the filter to the change detector can be formalized and realized as the marginalization of the above joint probability distribution with respect to $\mathbf{x}_k$

$$p(c_k^{ij} = \mathcal{C}) = \iiiint_{\mathbb{R}^4} p(\mathbf{x}_k, c_k^{ij} = \mathcal{C})\, d\mathbf{x}_k = \iiiint_{\mathbb{R}^4} p\left(c_k^{ij} = \mathcal{C} \mid \mathbf{x}_k\right) p\left(\mathbf{x}_k\right)\, d\mathbf{x}_k. \quad (16)$$

By using as prior on the state $p(\mathbf{x}_k)$ the RBE prediction $p^-(\mathbf{x}_k)$, this equation provides a way to let the current estimation of the state computed by the RBE module influence the prior for the Bayesian change detection algorithm, thereby realizing the cognitive feedback. Given the model from Section III-C, we can obtain an exact solution for (16).

By partitioning $\mathbb{R}^4$ into the two complementary subregions B_{ij} and $\bar{B}_{ij} = \mathbb{R}^4 \setminus B_{ij}$ of bounding boxes that contain or not the considered pixel (i, j), we obtain

$$\begin{aligned} p(c_k^{ij} = \mathcal{C}) &= K_1 \iiiint_{B_{ij}} p^-(\mathbf{x}_k)\, d\mathbf{x}_k + K_2 \iiiint_{\bar{B}_{ij}} p^-(\mathbf{x}_k)\, d\mathbf{x}_k \\ &= K_1 \iiiint_{\mathbf{x}_k \in B_{ij}} p^-(\mathbf{x}_k)\, d\mathbf{x}_k + K_2 \iiiint_{\mathbf{x}_k \in \mathbb{R}^4} p^-(\mathbf{x}_k)\, d\mathbf{x}_k \\ &\quad - K_2 \iiiint_{\mathbf{x}_k \in B_{ij}} p^-(\mathbf{x}_k)\, d\mathbf{x}_k \\ &= K_2 + (K_1 - K_2) I_k^{ij} \quad (17) \end{aligned}$$

$$I_k^{ij} \doteq \iiiint_{B_{ij}} p^-(\mathbf{x}_k)\, d\mathbf{x}_k. \quad (18)$$

Since I_k^{ij} varies in $[0, 1]$, it follows that, as intuitively predictable, $p(c_k^{ij} = \mathcal{C})$ varies in $[K_2, K_1]$: if no bounding box contains the pixel, we expect a probability that the pixel is changed equal to K_2; if all the bounding boxes contain the pixel the probability is K_1; it is a weighted average otherwise.

This reasoning holds for any distribution $p^-(\mathbf{x}_k)$ we might have on the state vector. In the following sections, we detail the two most relevant cases of using as the RBE filter, the Kalman filter, and the PF.

A. Cognitive Feedback With the Kalman Filter

By using the representation of the bounding box defined in (7) and assuming i_k^L, j_k^T, i_k^R, j_k^B to be independent,[1] the integral becomes

$$\begin{aligned} I_k^{ij} &= \iiiint_{\substack{i_k^L \le i \le i_k^R \\ j_k^T \le j \le j_k^B}\}B_{ij}} p(i_k^L)\,p(i_k^R)\,p(j_k^T)\,p(j_k^B)\,di_k^L di_k^R dj_k^T dj_k^B \\ &= \int_{-\infty}^{i} p(i_k^L)\,di_k^L \int_{i}^{+\infty} p(i_k^R)\,di_k^R \int_{-\infty}^{j} p(j_k^T)\,dj_k^T \int_{j}^{+\infty} p(j_k^B)\,dj_k^B \\ &= F_{i_k^L}(i)\big[1 - F_{i_k^R}(i)\big]F_{j_k^T}(j)\big[1 - F_{j_k^B}(j)\big] \end{aligned} \tag{19}$$

where F_x stands for the cumulative density function (CDF) of the random variable x.

Given that in the Kalman filter case all the PDFs are Gaussians, we can define all the factors of the product in (19) in terms of the standard Gaussian CDF, $\Phi(\cdot)$

$$\begin{aligned} I_k^{ij} = \Phi\left(\frac{i-\mu_{i_k^L}}{\sigma_{i_k^L}}\right)\Phi\left(\frac{\mu_{i_k^R}-i}{\sigma_{i_k^R}}\right) \\ \Phi\left(\frac{j-\mu_{j_k^T}}{\sigma_{j_k^T}}\right)\Phi\left(\frac{\mu_{j_k^B}-j}{\sigma_{j_k^B}}\right) \end{aligned} \tag{20}$$

where μ_x and σ_x are the mean and standard deviation of the random variable x. The factors of the product in (20) can be computed efficiently with only four searches in a precomputed lookup table of $\Phi(\cdot)$.

B. Cognitive Feedback With the PF

Despite the wider applicability of the PF with respect to the Kalman filter, the substitution of $p^-(\mathbf{x}_k)$ from (4) in (16) for the computation of the integral I_k^{ij} leads to the following simple formula:

$$I_k^{ij} = \sum_{n \in \mathcal{N}_k^{ij}} w_{k-1}^{(n)}, \quad \mathcal{N}_k^{ij} = \{(i,j) \in B(\mathbf{x}_k^{(n)})\}_{n=1}^{N} \tag{21}$$

that is, I_k^{ij} is obtained by summing up the weights of all the particles (bounding boxes) containing pixel (i, j). At each frame k, the $w \times h$ matrix $[I_k^{ij}]$ can be computed efficiently by scanning all the particles and incrementing by the current particle weight the values I_k^{ij} for all pixels (i, j) contained in the corresponding bounding-box.

V. Mining the Change Detection Output

In this section, we address the other side of the loop, i.e., the computation of a likelihood from the change detection output. Unfortunately, it is not possible for both filters to consider the change map as the current measurement $\mathbf{z}_k$ and derive a common model: the Kalman filter assumes a linear model while the relationship between the change map and the tracker state is clearly nonlinear. Hence, in Section V-A, we derive bottom-up the most likely bounding box from the change map and use it as the current measurement for the Kalman filter. Contrary to standard blob analysis-based methods, we do not threshold the change map and we can also estimate the time-varying measurement uncertainty (covariance matrix) $\mathbf{R}_k$; in Section V-B, instead, we can use as measurement $\mathbf{z}_k$ the change map $\mathbf{p}_k$ and define a proper likelihood $p(\mathbf{z}_k \mid \mathbf{x}_k)$ for the PF.

[1]Please note that this is a sensible assumption if we do not use a camera precalibrated with respect to the ground plane.

A. Measurements for the Kalman Filter

Given the change map $\mathbf{p}_k = [p(c_k^{ij} = \mathcal{C} \mid b_k^{ij}, f_k^{ij})]$ obtained by the Bayesian change detection algorithm in (9), our purpose is to compute the measurement $\mathbf{z}_k$ and its covariance $\mathbf{R}_k$ [42]. We use a time-invariant measurement matrix $\mathbf{H} = I_{4\times4}$. Hence, the measurement $\mathbf{z}_k$ lives in the same state-space of $\mathbf{x}_k$ (i.e., it represents a bounding box, to be computed from the change map at frame k). First, we note that

$$\begin{aligned} p(\mathbf{x}_k) &= \sum_{\mathbf{c}_k \in \Theta} p(\mathbf{x}_k, \mathbf{c}_k) \\ &= \sum_{\mathbf{c}_k \in \Theta} p(\mathbf{x}_k \mid \mathbf{c}_k)p(\mathbf{c}_k) \\ &= \sum_{\mathbf{c}_k \in \Theta} p(\mathbf{x}_k \mid \mathbf{c}_k)\prod_{ij} p(c_k^{ij}) \end{aligned} \tag{22}$$

where the last equality follows from the assumption of conditional independence between the categorical random variables c_k^{ij} providing the posterior change map computed by the Bayesian change detection. By using as change map $p(c_{ij}^k)$, the output of the change detection $\mathbf{p}_k$, (22) allows us to obtain the PDF of the bounding boxes for the current frame. From it, we can compute $\mathbf{z}_k$ and its uncertainty.

To use (22), we need an expression for the conditional probability $p(\mathbf{x}_k \mid \mathbf{c}_k)$ of the state given a change mask, based on the model for the ideal change map given the RBE state $p(\mathbf{c}_k \mid \mathbf{x}_k)$. Informally, this might be thought of as inverting the model in (14) and (15). After some manipulations, reported in Appendix A, we obtain

$$\begin{aligned} p(\mathbf{x}_k \mid \mathbf{c}_k) \propto \prod_{c_k^{ij}=\mathcal{C}} \frac{p(c_k^{ij} = \mathcal{C} \mid \mathbf{x}_k)}{K_C} \prod_{c_k^{ij}=\mathcal{U}} \frac{p(c_k^{ij} = \mathcal{U} \mid \mathbf{x}_k)}{1 - K_C} \\ K_C = K_2 + (K_1 - K_2)\left(\frac{1}{2}\right)^4. \end{aligned} \tag{23}$$

Then, by plugging (23) into (22), we obtain

$$\begin{aligned} p(\mathbf{x}_k) \propto \sum_{\mathbf{c}_k \in \Theta} \prod_{c_k^{ij}=\mathcal{C}} \frac{p(c_k^{ij} = \mathcal{C} \mid \mathbf{x}_k)p_k^{ij}}{K_C} \\ \prod_{c_k^{ij}=\mathcal{U}} \frac{p(c_k^{ij} = \mathcal{U} \mid \mathbf{x}_k)(1 - p_k^{ij})}{1 - K_C} \end{aligned} \tag{24}$$

where we use $\mathbf{p}_k$ for $p(\mathbf{c}_k)$ as mentioned. Noticing that we can express the resulting sum of products as a product of sums, as detailed in Appendix B, the previous formula can be rewritten as

$$p(\mathbf{x}_k) \propto \prod_{i,j}\left(\frac{p(c_k^{ij}=\mathcal{C} \mid \mathbf{x}_k)p_k^{ij}}{K_C} + \frac{p(c_k^{ij}=\mathcal{U} \mid \mathbf{x}_k)(1 - p_k^{ij})}{1 - K_C}\right). \tag{25}$$

By taking the logarithm of both sides, after some algebraic manipulations where we also exploit the fact that

$$\sum_{(i,j)\in\bar{B}(\mathbf{x}_k)} p_k^{ij} = \sum_{(i,j)} p_k^{ij} - \sum_{(i,j)\in B(\mathbf{x}_k)} p_k^{ij} \tag{26}$$

we attain

$$\ln p(\mathbf{x}_k) = \sum_{(i,j)\in B(\mathbf{x}_k)} \ln \frac{p_k^{ij} K_3 + K_4}{p_k^{ij} K_5 + K_6} + \gamma \tag{27}$$

where

$$\begin{aligned} K_3 &= K_1 - K_C \qquad K_4 = K_C(1-K_1) \\ K_5 &= K_2 - K_C \qquad K_6 = K_C(1-K_2) \end{aligned} \tag{28}$$

are constants and

$$\gamma = \ln \frac{1}{\alpha\left[K_C(1-K_C)\right]^{wh}} + \sum_{i,j} \ln\left[p_k^{ij} K_5 + K_6\right] \tag{29}$$

is an additive constant for a given frame, whose value turns out irrelevant for the sake of subsequent computations.

By letting $\mathbf{x}_k$ span, the space of all possible bounding boxes, (27) allows us to compute a nonparametric estimation of the log-PDF of a bounding box given the current change map. This holds independently of the PDF of the state.

In the Kalman filter, the PDF of the state vector (i_k^b, j_k^b, w_k, h_k) is Gaussian. In such a case, the variables $(i_k^L, j_k^T, i_k^R, j_k^B)$ are a linear combination of Gaussian random variables. In addition, we are assuming that variables $(i_k^L, j_k^T, i_k^R, j_k^B)$ are independent (Section IV). Therefore, $(i_k^L, j_k^T, i_k^R, j_k^B)$ are jointly Gaussian and the mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$ of the state is fully defined by the four means μ_k^L, μ_k^R, μ_k^T, μ_k^B and the four standard deviations σ_k^L, σ_k^R, σ_k^T, σ_k^B.

Then, an estimate $\widehat{\boldsymbol{\mu}}_k$ of the mean of the state vector $\boldsymbol{\mu}_k$ can be obtained by observing that, due to the logarithm being a monotonically increasing function, the mode of the computed log-PDF coincides with the mode of the PDF, and that, due to the Gaussianity assumption, the mode of the PDF coincides with its mean. Hence, we obtain an estimate $\widehat{\boldsymbol{\mu}}_k$ of $\boldsymbol{\mu}_k$ by searching for the bounding box that maximizes the right-hand side of (27)

$$\begin{aligned} \widehat{\boldsymbol{\mu}}_k &= \operatorname*{argmax}_{\mathbf{x}} h(\mathbf{x}, \mathbf{p}_k) \\ &\doteq \operatorname*{argmax}_{\mathbf{x}} \sum_{(i,j)\in B(\mathbf{x})} \ln \frac{p_k^{ij} K_3 + K_4}{p_k^{ij} K_5 + K_6}. \end{aligned} \tag{30}$$

To compute the standard deviations, first, we substitute the expression of the Gaussian PDF to $p(\mathbf{x}_k)$ in the left-hand side of (27) as

$$\begin{aligned} &\gamma - 2\ln(2\pi) - \ln\left(\sigma_k^L \sigma_k^R \sigma_k^T \sigma_k^B\right) - \frac{\left(i_k^L - \mu_k^L\right)^2}{2\sigma_k^{L\,2}} \\ &- \frac{\left(i_k^R - \mu_k^R\right)^2}{2\sigma_k^{R\,2}} - \frac{\left(j_k^T - \mu_k^T\right)^2}{2\sigma_k^{T\,2}} - \frac{\left(j_k^B - \mu_k^B\right)^2}{2\sigma_k^{B\,2}} = h\left(\mathbf{x}_k, \mathbf{p}_k\right). \end{aligned} \tag{31}$$

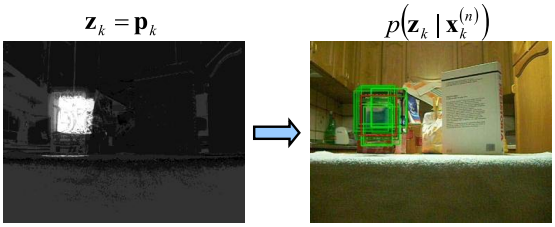

Fig. 4. Input (left) and output (right) of the observation likelihoods module. Left: brighter the pixels, the greater the posterior probability to belong to the tracked object computed by the change detector. Right: brighter the bounding boxes, the greater the likelihood for the corresponding particles. The particle with the highest weight is marked in red.

Then, we impose that (31) is satisfied at the estimated mean point $\widehat{\boldsymbol{\mu}}$ and that all the standard deviations are equal, i.e., $\sigma_k^L = \sigma_k^R = \sigma_k^T = \sigma_k^B = \sigma_k$, thus achieving

$$\gamma = 2\ln(2\pi) + 2\ln\sigma_k^2 + h(\widehat{\boldsymbol{\mu}}_k, \mathbf{p}_k). \tag{32}$$

By substituting in (31), the above expression for γ and the estimated $\widehat{\boldsymbol{\mu}}_k$ for $\boldsymbol{\mu}_k$, we can compute an estimate $\widehat{\sigma}_k^2(\mathbf{x})$ of the variance σ_k^2 by imposing (31) for whatever bounding box $\mathbf{x} \neq \widehat{\boldsymbol{\mu}}_k$. In particular, we obtain

$$\widehat{\sigma}_k^2(\mathbf{x}) = \frac{1}{2} \frac{\left\|\widehat{\boldsymbol{\mu}}_k - \mathbf{x}\right\|_2^2}{h(\widehat{\boldsymbol{\mu}}_k, \mathbf{p}_k) - h(\mathbf{x}, \mathbf{p}_k)}. \tag{33}$$

To obtain a more robust estimate, we average $\widehat{\sigma}_k^2(\mathbf{x})$ over a neighborhood of the estimated mean bounding box $\widehat{\boldsymbol{\mu}}_k$. Finally, to compute the means and covariance of the measurements for the Kalman filter, we exploit the properties of linear combinations of Gaussian variables

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^{-1} \end{bmatrix} \widehat{\boldsymbol{\mu}}_k \tag{34}$$

$$\mathbf{R}_k = \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^{-1} \end{bmatrix}^T \widehat{\sigma}_k^2. \tag{35}$$

Such variables can then be used in the Kalman filter update formulas [42], to calculate the updated tracker state and close the proposed Bayesian loop.

B. Likelihood for the PF

The purpose of this module is to compute the observation likelihood $p(\mathbf{z}_k \mid \mathbf{x} = \mathbf{x}_k^{(n)})$ for each particle given the change map $\mathbf{p}_k = [p(c_k^{ij} = \mathcal{C} \mid \mathbf{b}_k, \mathbf{f}_k)]$ provided by the Bayesian change detector in (9), which is taken as the current observation $\mathbf{z}_k$. These likelihoods will then be plugged into the particle weights updating formula (5) so as to define the state posterior (3) and close the loop. In Fig. 4, we show visually the input/output of this module for the same sample frame as in Fig. 3.

To compute the likelihoods $p\left(\mathbf{z}_k|\mathbf{x}=\mathbf{x}_k^{(n)}\right) \doteq p(\mathbf{p}_k|\mathbf{x}=\mathbf{x}_k^{(n)})$, we follow the common approach (see [26]) to define the likelihood as a similarity score between the current observation and a reference model. In our case, (15) defines such a reference change map. Hence, the similarity score must compare two change maps, i.e., two matrices of Bernoulli distributions. To this aim, we employ the average over all the pixels (i, j)

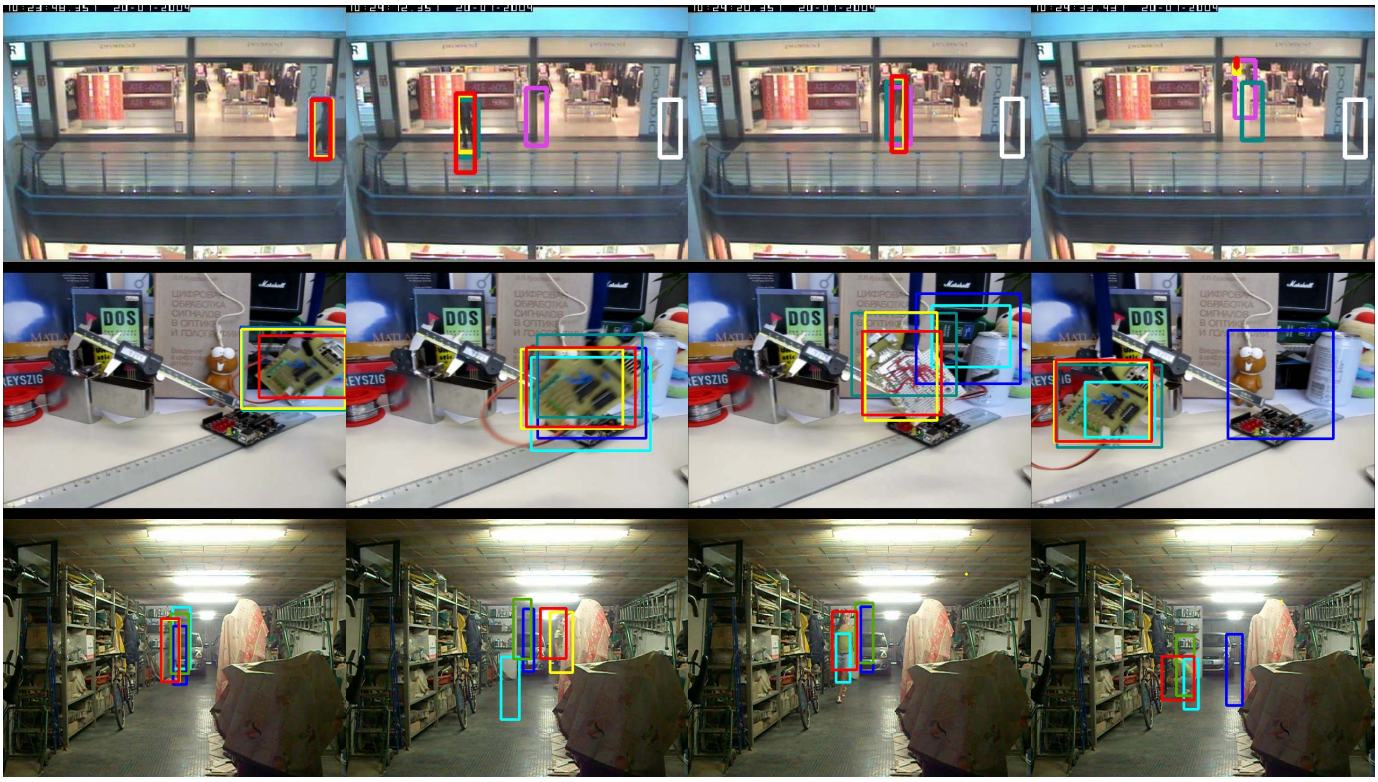

Fig. 5. Screenshots from a sequence of every data sets. From top to bottom: CAVIAR2, PROST2, and BLOCK6. For the sake of readability, we have reported, besides our Bayesian loops, only the three best performing trackers for each sequence. Trackers: PBL (red), KBL (yellow), FT (dark green), STRUCK (dark cyan), SemiBoost (white), MS (cyan), PF (blue), and MILBoost (pink).

of the Bhattacharyya coefficients BC_k^{ij} between the Bernoulli distributions in the observation $\mathbf{p}_k$ and the corresponding ones in the reference model $p(\mathbf{c}_k \,|\, \mathbf{x} = \mathbf{x}_k^{(n)})$ as

$$p\left(\mathbf{z}_k \,|\, \mathbf{x}_k^{(n)}\right) = \frac{1}{wh} \sum_{i,j} BC_k^{ij} \tag{36}$$

$$= \frac{1}{wh} \sum_{i,j} \left[\sqrt{p_k^{ij}\; p\left(c_k^{ij} = \mathcal{C} \,|\, \mathbf{x} = \mathbf{x}_k^{(n)}\right)} + \sqrt{\left(1 - p_k^{ij}\right) p\left(c_k^{ij} = \mathcal{U} \,|\, \mathbf{x} = \mathbf{x}_k^{(n)}\right)} \right]. \tag{37}$$

Actually, the likelihoods $p\left(\mathbf{z}_k \,|\, \mathbf{x}_k^{(n)}\right)$ can be defined up to a proportionality constant, since the weights they contribute to update in (5) are then normalized to sum up to 1. Hence, we can drop the factor $1/wh$. In addition, partitioning the image into the rectangular region $B(\mathbf{x}_k^{(n)})$ delimited by the bounding box $\mathbf{x}_k^{(n)}$ and its complement, by using (26) and substituting the model (15) in (37), we obtain

$$p\left(\mathbf{z}_k \,|\, \mathbf{x}_k^{(n)}\right) = K_7 \sum_{B\left(\mathbf{x}_k^{(n)}\right)} \sqrt{p_k^{ij}} + K_8 \sum_{B\left(\mathbf{x}_k^{(n)}\right)} \sqrt{1 - p_k^{ij}} + K_9 \tag{38}$$

where

$$K_7 = \sqrt{K_1} - \sqrt{K_2}$$
$$K_8 = \sqrt{1 - K_1} - \sqrt{1 - K_2} \tag{39}$$

are constants and though

$$K_9 = \sqrt{K_2} \sum_{i,j} \sqrt{p_k^{ij}} + \sqrt{1 - K_2} \sum_{i,j} \sqrt{1 - p_k^{ij}} \tag{40}$$

can vary from one frame to another due to variations of the observation $\mathbf{p}_k$, it is constant in a given frame. Hence, the likelihoods in (38) can be computed very efficiently by using two integral images [46] accumulating $(p_k^{ij})^{(1/2)}$ and $(1 - p_k^{ij})^{(1/2)}$, respectively. In particular, the algorithm has linear complexity in the number of particles and constant complexity in the size of the bounding boxes.

VI. Experimental Results

In this section, we present two sets of experiments. The first compares our proposals to several well-known trackers on publicly available data sets, the second aims at assessing performance on a novel challenging data set, which will be made publicly available.

We evaluate performance based on the overlap between the estimated tracking result and the ground truth. We quantify the amount of overlap using the Dice [47] in each frame k, which is defined as

$$d_k = \frac{2\left|B(\mathbf{x}_k) \cap B(\mathbf{x}_k^{GT})\right|}{|B(\mathbf{x}_k)| + \left|B(\mathbf{x}_k^{GT})\right|} \tag{41}$$

with $|.|$ denoting the cardinality of a set; $d_k \in [0, 1]$ and $d_k = 1$ in case of perfect overlap. In addition, in Fig. 5 and the videos contained in the supplementary material, we provide

TABLE II

DICE OVERLAP FOR ALL SEQUENCES. STOCHASTIC TRACKERS HAVE BEEN RUN 10 TIMES AND THE MEAN VALUE IS REPORTED. PBL, KBL, MS, COLOR-BASED PF, IVT, FT, SCM. THE BEST TRACKER IN EACH SEQUENCE IS HIGHLIGHTED IN BOLD, THE SECOND BEST IS UNDERLINED. THE SCORE OF TRACKERS THAT LOSE THE TARGET IS IN RED

	Using background				Without background									
	Stochastic		Deterministic		Stochastic								Deterministic	
Sequence	PBL	PF [26]	KBL	MS [13]	PF [26]	IVT [30]	Boost [36]	SemiB. [37]	MILB. [38]	STRUCK [43]	TLD [44]	SCM [35]	MS [13]	FT [27]
CAVIAR 1	0.70	0.01	**0.74**	0.05	0.02	0.30	0.16	0.20	0.48	0.63	0.02	0.55	0.50	0.55
CAVIAR 2	0.64	0.01	**0.66**	0.01	0.01	0.01	0.01	0.02	0.35	0.41	0.01	0.10	0.01	0.01
CAVIAR 3	0.69	0.01	**0.70**	0.05	0.01	0.02	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0.02
PROST 1	**0.69**	0.03	**0.69**	0.62	0.30	0.58	0.16	0.35	0.18	0.59	0.06	0.37	0.45	0.11
PROST 2	0.74	0.02	0.72	0.46	0.61	0.60	0.16	0.09	0.55	**0.82**	0.43	0.50	0.64	0.47
PROST 3	0.72	0.04	**0.74**	0.67	0.67	0.32	0.07	0.02	0.44	0.66	0.56	0.32	0.57	0.10
BLOCK 1	**0.60**	0.03	0.08	0.30	0.56	0.20	0.12	0.21	0.06	0.10	0.27	0.29	0.28	0.15
BLOCK 2	**0.71**	0.02	0.22	0.13	0.22	0.09	0.03	0.10	0.38	0.38	0.25	0.08	0.01	0.05
BLOCK 3	**0.67**	0.02	0.45	0.10	0.14	0.14	0.16	0.14	0.17	0.18	0.14	0.42	0.08	0.13
BLOCK 4	0.72	0.05	**0.75**	0.19	0.07	0.05	0.01	0.42	0.37	0.51	0.0	0.22	0.20	0.11
BLOCK 5	**0.69**	0.06	0.24	0.19	0.19	0.34	0.16	0.08	0.25	0.5	0.15	0.14	0.22	0.19
BLOCK 6	**0.56**	0.01	0.17	0.24	0.33	0.07	0.04	0.02	0.26	0.16	0.18	0.21	0.29	0.43

qualitative results showing the bounding boxes yielded by the trackers.[2] In particular, for the sake of clarity, we report the bounding boxes yielded by our Bayesian loops and by the three best performing trackers among the others.

To benchmark our approach, we consider standard solutions for visual tracking, such as the mean-shift tracker [13], the PF based on color histograms [26] and the integral histogram tracker FragTrack (FT) [27], as well as several recent trackers based on the idea of adaptive target modeling: Boost [36], SemiBoost [37], incremental visual tracker (IVT) [30], MILBoost [38], STRUCK [43], TLD [44], and sparse collaborative model (SCM) [35].[3] All these trackers do not explicitly exploit the presence of a static camera. Therefore, we also include in the comparison the less known variant of mean-shift and the color-based PF, presented in [13] and [26], that modify the target histogram and the observation likelihood to take advantage explicitly of this configuration. As for the considered adaptive trackers, none defines an explicit variant for tracking with static cameras. However, all but IVT [30] are discriminative trackers, i.e., they approach the tracking problem in terms of continuous classification of patches extracted from the current frame as either target or background: as such, they should benefit significantly from the static camera scenario.

All trackers are initialized with the same input bounding box in each sequence, taken from the ground truth. As for parameters, we use the values reported in the original papers or in the available implementations. The parameters K_1 and K_2 of our Bayesian Loop were coarsely tuned on a sequence not considered in the evaluation (i.e., $K_1 = 0.6$, $K_2 = 0.4$), and then kept constant throughout all sequences and for both the Kalman as well as the PF formulation. Concerning the Kalman filter parameters, we used a constant velocity motion model, with $\sigma^2_{i^b} = \sigma^2_{j^b} = \sigma^2_w = \sigma^2_h = 1$ and $\sigma^2_{v^b_i} = \sigma^2_{v^b_j} = 10$, with v^b_i and v^b_j the velocity components. Concerning the PF parameters, we used 5000 particles and a zero-mean Gaussian transition model, with $\sigma^2_{i^b} = \sigma^2_{j^b} = 10$ and $\sigma^2_w = \sigma^2_h = 3$.

A. Publicly Available Data Sets

We use three sequences from the CAVIAR data set[4] and three from the PROST data set.[5] They are all rather long sequences (about 900 frames on average) presenting a variety of nuisances: out-of-plane rotations, occlusions, scale changes, motion blur, small and untextured targets, deformable targets, cluttered background, and appearance changes.

Results are reported in the first six rows of Table II. The first remark is that the proposed Bayesian loop, either based on Kalman [Kalman filter Bayesian loop (KBL)] or particle [Particle filter Bayesian loop (PBL)] filtering, clearly outperforms all other trackers: our proposal turns out always the first or second best in every sequence but PROST2, where STRUCK obtains the best overlap, followed by the proposed Bayesian loops. The table suggests also that CAVIAR is more challenging than PROST for the considered trackers, as vouched by the lower number of algorithms that keep tracking the target until the end of the sequences as well as by the low Dice score yielded by those, such as MILBoost and STRUCK, that can do so. Indeed, in the typical video surveillance scenario addressed by the CAVIAR data set, which deploys wide angle static cameras, targets turn out too untextured and small for their models to be discriminative (see the first row in Fig. 5). Even the inclusion of background in PF and mean shift (MS) (2nd and 4th column of Table II, respectively) does not allow to counteract such low discriminative power of the target model.

On the other hand, the PROST data set, although featuring short occlusions and fast motion changes, deals with wider and more textured targets, which accounts for the overall better performance of algorithms. The performance of MS is particularly interesting: when the target is rectangular and aligned with image axes (PROST1 and 3), down-weighting the background colors in the target model as done by MS is beneficial (compare 4th with 12th column), it is detrimental

[2]For each sequence, we plot the bounding boxes of the run closest to the mean performance of each tracker, i.e., the run that minimizes the $l1$ norm of the Dice with respect to the mean run [39].

[3]Although we use the PROST data set, we do not consider PROST [48] in the evaluation as the authors do not provide a reference implementation.

[4]Data coming from the EC Funded CAVIAR project/IST 2001 37540, found at URL: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

[5]http://gpu4vision.icg.tugraz.at/index.php?content=subsites/prost/prost.php

Fig. 6. From left to right: exemplar frame from the BLOCK2 sequence (shown in grayscale as deployed by the Bayesian Loop); resulting change map; original color frame with overlaid the 20 particles with the highest weights (in red the highest one, perfectly enclosing the target).

otherwise (PROST2, e.g., third frame on second row of Fig. 5). The proposed Bayesian loops, instead, are not affected by the misalignment of the target, obtaining the second and third best performance on PROST2. This shows that, despite the assumption of a rectangular target, the proposed framework has a higher adaptability than standard trackers based on the same geometrical model. Finally, the overall performance of the two variants of the proposed Bayesian loop is roughly equivalent on publicly available data sets.

B. BLOCK Data Set

To assess the robustness of trackers in even more challenging settings as well as to highlight the diverse behavior of the two variants of our Bayesian Loop, we introduce a novel data set. In particular, the six new videos comprise a sequence (BLOCK6), illustrated in the last row of Fig. 5, addressing a similar scenario as in the CAVIAR data set, but with occlusions, heavier scale, and form factor changes. Four videos (BLOCK1, 2, 3, and 5) somehow resemble those included in PROST, i.e., an object attached to a rope is moved in front of a cluttered background, but with more prolonged and more frequent occlusions as well as faster movements. Finally, one video (BLOCK4) presents a sudden inversion of motion direction during occlusion.

Results are reported in the last six rows of Table II and the last row of Fig. 5. The gap between PBL and other previous proposals becomes even more evident than on public data sets. In addition, in these sequences PBL neatly outperforms also KBL. Indeed, all described nuisances cannot be effectively dealt with by a tracker that does not take into account possibly multimodal posteriors and nonlinear motion patterns. For instance, in the third row of Fig. 5, PBL is the only tracker that never loses the target, whereas KBL loses it after the occlusion depicted in the second frame.

The proposed communication flows between RBE and change detection are another ingredient of the success of the proposed method. As it can be seen in Fig. 6, the cognitive feedback guides the change detector to assign high probabilities to be changed to the target region of the image, in particular also to those pixels that would otherwise result more likely unchanged due to camouflage. Similarly, Fig. 6 shows that principled model-based definition of the likelihood allows for dealing effectively and effortlessly with change maps that would be thresholded into multiple blobs and hinder heuristically designed methods.

C. Run Times

As a final remark, we would like to emphasize that the notable performance of our Bayesian loops comes without sacrificing efficiency: even with a high number of particles, such as 5000, the PF loop runs at 20 FPS without resorting to multithreading or GPU optimizations, although parallelism could be easily exploited. This is not a secondary feature for an algorithm that deploys particle filtering, whose robustness usually comes at the expense of real-time processing. As for the Kalman filter loop, it runs comfortably on off-the-shelf hardware at 25 FPS. All the tests were performed on a PC equipped with an Intel Core i7 3 GHz and 6 GB of RAM, running MS Windows 7 Professional.

VII. Conclusion

The problem of visual tracking, even in the simplified though widespread scenario of static cameras, is yet to be solved. In this paper, we have shown that significant advances can be reached by creating a principled communication flow between a recursive Bayesian estimator, such as the Kalman or PF, and a Bayesian change detector. Accordingly, an observation likelihood for the recursive filter is derived from the change map without any hard-thresholding, while a cognitive feedback from the recursive filter steers the change detector. The proposed theoretical formulation relies on a remarkably simple model of the ideal change map given the target state, which encompasses the only two parameters of the overall computation.

Several interesting extensions can be envisaged. The first issue we plan to investigate is how to formulate both communication flows to address multitarget tracking scenarios; then, it would be useful to derive a sound communication also for more general models of the ideal change map, e.g., for nonrectangular targets: this could be beneficial for tracking highly deformable targets, such as limbs and hands in natural human–computer interfaces; finally, the inclusion of foreground appearance into the model could improve its discriminative power, thus fostering extension to multiple targets.

Appendix A
Derivation of (23)

By Bayes rule, (14) and independence of the variables c_{ij}

$$p(\mathbf{x}_k \mid \mathbf{c}_k) = \hat{p}(\mathbf{x}_k)\frac{p(\mathbf{c}_k \mid \mathbf{x}_k)}{\hat{p}(\mathbf{c}_k)} = \hat{p}(\mathbf{x}_k)\prod_{i,j}\frac{p\left(c_k^{ij} \mid \mathbf{x}_k\right)}{\hat{p}\left(c_k^{ij}\right)}. \quad (42)$$

We have used the notation $\hat{p}(\mathbf{x}_k)$ and $\hat{p}(c_k^{ij})$ in (42) since here these PDFs must be interpreted differently than in (22): in (22), $p(\mathbf{x}_k)$ and $p(c_k^{ij})$ represent, respectively, the PDF of the measurement and the change map of the current frame, while in (42), both must be interpreted as priors that form part of our model for $p(\mathbf{x}_k \mid \mathbf{c}_k)$, which is independent of the current frame. Furthermore, using as prior on the state $\hat{p}(\mathbf{x}_k)$ the prediction of the RBE filter, as done in the cognitive feedback section, would have created a strong coupling between the output of the sensor and the previous state of the filter, that does not fit the RBE framework, where measurements depend on the current state only, and could easily lead the loop to diverge. Hence, we assume a uniform noninformative prior $\hat{p}(\mathbf{x}_k) = 1/\alpha$ for the state.

The analysis conducted for the cognitive feedback is useful to expand each $\hat{p}(c_k^{ij})$ in (42). Since we are assuming a uniform prior on an infinite domain for the state variables, i.e., a symmetric PDF with respect to $x = 0$, it turns out that its CDF is constant and equals to $1/2$

$$\mathrm{CDF}(x) = \frac{1}{\alpha}x + \frac{1}{2} \xrightarrow{\alpha \to +\infty} \frac{1}{2}. \tag{43}$$

Hence, every $\hat{p}(c_k^{ij})$ in (42) can be expressed using (17) and (19) as

$$\hat{p}\big(c_k^{ij} = \mathcal{C}\big) = K_2 + (K_1 - K_2)\Big(\frac{1}{2}\Big)^4 \doteq K_C. \tag{44}$$

Therefore, the final inverse model is

$$p(\mathbf{x}_k \mid \mathbf{c}_k) = \frac{1}{\alpha} \prod_{i,j:c_k^{ij}=\mathcal{C}} \frac{p\big(c_k^{ij} = \mathcal{C} \mid \mathbf{x}_k\big)\, p\big(c_k^{ij} = \mathcal{C}\big)}{K_C} \prod_{i,j:c_k^{ij}=\mathcal{U}} \frac{p(c_k^{ij} = \mathcal{U} \mid \mathbf{x}_k)\, p\big(c_k^{ij} = \mathcal{U}\big)}{1 - K_C}. \tag{45}$$

Appendix B
Derivation of (25)

We start from

$$p(\mathbf{x}_k) \propto \sum_{\mathbf{c}_k \in \Theta} \prod_{c_k^{ij}=\mathcal{C}} \frac{p\big(c_k^{ij} = \mathcal{C} \mid \mathbf{x}_k\big)\, p_k^{ij}}{K_C} \prod_{c_k^{ij}=\mathcal{U}} \frac{p\big(c_k^{ij} = \mathcal{U} \mid \mathbf{x}_k\big)\big(1 - p_k^{ij}\big)}{1 - K_C} \tag{46}$$

which represents a sum over all possible change masks. If, for instance, our image is 3 by 1, the set of all possible change masks $\Theta = \{\mathcal{C}, \mathcal{U}\}^{3\times 1}$ is

$$\Theta = \{\{\mathcal{CCC}\}, \{\mathcal{CCU}\}, \{\mathcal{CUC}\}, \{\mathcal{CUU}\} \{\mathcal{UCC}\}, \{\mathcal{UCU}\}, \{\mathcal{UUC}\}, \{\mathcal{UUU}\}\}. \tag{47}$$

As can be noted from the toy example, there are half change masks where $c_k^{11} = \mathcal{C}$ and half where $c_k^{11} = \mathcal{U}$. The remaining 2 by 1 submasks are identical in both subsets. We can collect the common factor $p(c_k^{11} = \mathcal{C} \mid \mathbf{x}_k)\, p_k^{11}/K_C$ among the first half of addends and the common factor $p(c_k^{11} = \mathcal{U} \mid \mathbf{x}_k)(1 - p_k^{11})/(1 - K_C)$ among the second half. If we explicit this in (46), we obtain

$$p(\mathbf{x}_k) \propto \frac{p\big(c_k^{11} = \mathcal{C} \mid \mathbf{x}_k\big)\, p_k^{11}}{K_C} p_{\Theta\setminus 11}(\mathbf{x}_k) + \frac{p\big(c_k^{11} = \mathcal{U} \mid \mathbf{x}_k\big)\big(1 - p_k^{11}\big)}{1 - K_C} p_{\Theta\setminus 11}(\mathbf{x}_k) \tag{48}$$

$$= p_{\Theta\setminus 11}(\mathbf{x}_k)\left(\frac{p\big(c_k^{11} = \mathcal{C} \mid \mathbf{x}_k\big)\, p_k^{11}}{K_C} + \frac{p(c_k^{11} = \mathcal{U} \mid \mathbf{x}_k)(1 - p_k^{11})}{1 - K_C}\right) \tag{49}$$

where $p_{\Theta\setminus 11}(\mathbf{x}_k)$ is the PDF estimated without pixel $(1, 1)$

$$p_{\Theta\setminus 11}(\mathbf{x}_k) \propto \sum_{\mathbf{c}_k \in \Theta\setminus 11} \prod_{c_k^{ij}=\mathcal{C}} \frac{p\big(c_k^{ij} = \mathcal{C} \mid \mathbf{x}_k\big)\, p_k^{ij}}{K_C} \prod_{c_k^{ij}=\mathcal{U}} \frac{p\big(c_k^{ij} = \mathcal{U} \mid \mathbf{x}_k\big)\big(1 - p_k^{ij}\big)}{1 - K_C} \tag{50}$$

that is, the same addends of (46) but with one pixel less. Therefore, we can extract pixel $(2, 1)$ from it as we have done for pixel $(1, 1)$ from (46), to obtain

$$p(\mathbf{x}_k) \propto p_{\Theta\setminus\{21,11\}}(\mathbf{x}_k) \left(\frac{p\big(c_k^{21} = \mathcal{C} \mid \mathbf{x}_k\big)\, p_k^{21}}{K_C} + \frac{p\big(c_k^{21} = \mathcal{U} \mid \mathbf{x}_k\big)\big(1 - p_k^{21}\big)}{1 - K_C}\right) \left(\frac{p(c_k^{11} = \mathcal{C} \mid \mathbf{x}_k)\, p_k^{11}}{K_C} + \frac{p\big(c_k^{11} = \mathcal{U} \mid \mathbf{x}_k\big)\big(1 - p_k^{11}\big)}{1 - K_C}\right). \tag{51}$$

We can continue this line of reasoning until we get to the product of sums form presented in (25).

References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. 1–45, Dec. 2006.

[2] I. Haritaoglu, D. Harwood, and L. S. Davis, "W^4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[3] R. T. Collins, A. J. Lipton, and T. Kanade, "A system for video surveillance and monitoring," Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-00-12, 1999.

[4] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Washington, DC, USA, Jun. 1999, pp. 246–252.

[5] T. Zhao and R. Nevatia, "Tracking multiple humans in complex situations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1208–1221, Sep. 2004.

[6] M. Harville and D. Li, "Fast, integrated person tracking and activity recognition with plan-view templates from a single stereo camera," in *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Washington, DC, USA, Jul. 2004, pp. 398–405.

[7] M. Isard and J. MacCormick, "BraMBLe: A Bayesian multiple-blob tracker," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, vol. 2. Washington, DC, USA, Jul. 2001, pp. 34–41.

[8] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1805–1819, Nov. 2005.

[9] X. Song, J. Cui, H. Zha, and H. Zhao, "Vision-based multiple interacting targets tracking via on-line supervised learning," in *Proc. 10th Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 642–655.

[10] S. M. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 505–519, Mar. 2009.
[11] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using K-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.
[12] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
[13] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.
[14] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, and L. Van Gool, "Depth-from-recognition: Inferring metadata by cognitive feedback," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Washington, DC, USA, Oct. 2007, pp. 1–8.
[15] S. Salti, A. Lanza, and L. Di Stefano, "Bayesian loop for synergistic change detection and tracking," in *Proc. Int. Workshop Vis. Surveill. (VS)*, 2010, pp. 43–53.
[16] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed, "Moving object detection in spatial domain using background removal techniques—State-of-art," *Recent Patents Comput. Sci.*, vol. 1, no. 1, pp. 32–54, 2008.
[17] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Washington, DC, USA, 2000, pp. 751–767.
[18] N. Ohta, "A statistical approach to background subtraction for surveillance systems," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, vol. 2. Washington, DC, USA, Jul. 2001, pp. 481–486.
[19] J. Lou, H. Yang, W. Hu, and T. Tan, "An illumination-invariant change detection algorithm," in *Proc. 5th Asian Conf. Comput. Vis. (ACCV)*, vol. 1. Jan. 2002, pp. 13–18.
[20] B. Xie, V. Ramesh, and T. Boult, "Sudden illumination change detection using order consistency," *Image Vis. Comput.*, vol. 22, no. 2, pp. 117–125, Feb. 2004.
[21] A. Lanza and L. Di Stefano, "Statistical change detection by the pool adjacent violators algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1894–1910, Sep. 2011.
[22] A. Mittal and V. Ramesh, "An intensity-augmented ordinal measure for visual correspondence," in *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. Washington, DC, USA, Jun. 2006, pp. 849–856.
[23] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1305–1312.
[24] L. Taycher, J. W. Fisher, and T. Darrell, "Incorporating object tracking feedback into background maintenance framework," in *Proc. Workshop Motion Video Comput. (WACV/MOTION)*, vol. 2. Washington, DC, USA, Jan. 2005, pp. 120–125.
[25] M. Harville, "A framework for high-level feedback to adaptive, per-pixel, mixture-of-Gaussian background models," in *Proc. 7th Eur. Conf. Comput. Vis. (ECCV)*, Jul. 2002, pp. 543–560.
[26] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. 7th Eur. Conf. Comput. Vis. (ECCV)*, 2002, pp. 661–675.
[27] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. Washington, DC, USA, Jun. 2006, pp. 798–805.
[28] W. He, T. Yamashita, H. Lu, and S. Lao, "SURF tracking," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1586–1592.
[29] J. Jiang and A. Yilmaz, "Persistent tracking of static scene features using geometry," *Comput. Vis. Image Understand.*, vol. 120, pp. 141–156, Mar. 2014.
[30] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.
[31] Q. Yu, T. B. Dinh, and G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," in *Proc. 10th Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 678–691.
[32] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1631–1643, Oct. 2005.
[33] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, Jun. 2004.
[34] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1296–1311, Oct. 2003.
[35] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1838–1845.
[36] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. Washington, DC, USA, Jun. 2006, pp. 260–267.
[37] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. 10th Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 234–247.
[38] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.
[39] S. Salti, A. Cavallaro, and L. Di Stefano, "Adaptive appearance modeling for video tracking: Survey and evaluation," *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 4334–4348, Oct. 2012.
[40] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014, doi: 10.1109/TPAMI.2013.230.
[41] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2411–2418.
[42] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Fluids Eng.*, vol. 82, no. 1, pp. 35–45, 1960.
[43] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 263–270.
[44] Z. Kalal, J. Matas, and K. Mikolajczyk, "Online learning of robust object detectors during unstable tracking," in *Proc. 12th Int. Conf. Comput. Vis. (ICCV)*, Kyoto, Japan, Sep./Oct. 2009, pp. 1417–1424.
[45] A. Lanza, S. Salti, and L. Di Stefano, "Background subtraction by non-parametric probabilistic clustering," in *Proc. IEEE Int. Conf. AVSS*, Aug./Sep. 2011, pp. 243–248.
[46] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2002.
[47] A. T. Nghiem, F. Bremond, M. Thonnat, and V. Valentin, "ETISEO, performance evaluation for video surveillance systems," in *Proc. AVSS*, Sep. 2007, pp. 476–481.
[48] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: Parallel robust online simple tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 723–730.

Samuele Salti (M'09) received the M.Sc. and Ph.D. degrees in computer science engineering from the University of Bologna, Bologna, Italy, in 2007 and 2011, respectively.

He has held a post-doctoral position with the Computer Vision Laboratory, Department of Computer Science and Engineering, University of Bologna, since 2011. He visited the Heinrich-Hertz-Institute, Berlin, Germany, in 2007, where he was involved in human–computer interaction, and the Multimedia and Vision Research Group with the Queen Mary University of London, London, U.K., in 2010, where he was involved in adaptive appearance models for video tracking. He has co-authored 22 publications in international conferences and journals. His current research interests include adaptive video tracking, 3-D shape matching, Bayesian filtering, and object recognition.

Dr. Salti is a member of the Italian Group Researchers in Pattern Recognition. He serves as a reviewer of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and several international conferences. He was a recipient of the Best Paper Award Runner-Up at the International Conference on 3-D Imaging, Modeling, Processing, Visualization and Transmission in 2011.

Alessandro Lanza received the M.S. degree in civil engineering from the University of Pavia, Pavia, Italy, in 2000, and the European Ph.D. degree in information technology from the Advanced Research Centre on Electronic Systems for Information and Communication Technologies, University of Bologna, Bologna, Italy, in 2007.

He was a Doctoral Fellow with the École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2006, for six months. He is currently a Post-Doctoral Fellow with the Department of Mathematics, University of Bologna. His current research interests include computer vision, pattern recognition, and numerical analysis.

Luigi Di Stefano (M'04) received the M.S. degree in electronic engineering from the University of Bologna, Bologna, Italy, in 1989, and the Ph.D. degree in electronic engineering and computer science from the Department of Electronics, Computer Science and Systems, University of Bologna, in 1994.

He was a Post-Doctoral Fellow with Trinity College, Dublin, Ireland, in 1995, for six months. He is currently an Associate Professor with the Department of Electronics, Computer Science and Systems, University of Bologna. He has authored over 120 papers, and holds five patents. His current research interests include computer vision, image processing, and computer architecture.

Prof. Di Stefano is a member of the IEEE Computer Society and the International Association Pattern Recognition–Italian Chapter, and has been a member of the Scientific Advisory Board of the Datalogic Group since 2012.