Journal Computer Vision and Image Understanding 131 (2015): 42-55

DOI: http://dx.doi.org/10.1016/j.cviu.2014.06.016

Temporal validation of particle filters for video tracking☆

Juan C. SanMiguel[a,b,*], Andrea Cavallaro[b]

[a] *Video Processing and Understanding Lab, Universidad Autónoma of Madrid, Spain*
[b] *Centre for Intelligent Sensing, Queen Mary University of London, United Kingdom*

Abstract

We present an approach for determining the temporal consistency of particle filters in video tracking based on model validation of their uncertainty over sliding windows. The filter uncertainty is related to the consistency of the dispersion of the filter hypotheses in the state space. We learn an uncertainty model via a mixture of Gamma distributions whose optimum number is selected by modified information-based criteria. The time-accumulated model is estimated as the sequential convolution of the uncertainty model. Model validation is performed by verifying whether the output of the filter belongs to the convolution model through its approximated cumulative density function. Experimental results and comparisons show that the proposed approach improves both precision and recall of competitive approaches such as Gaussian-based online model extraction, bank of Kalman filters and empirical thresholding. We combine the proposed approach with a state-of-the-art online performance estimator for video tracking and show that it improves accuracy compared to the same estimator with manually tuned thresholds while reducing the overall computational cost.

Keywords: Particle filter, Uncertainty, Model validation, Change detection, Performance evaluation, Video tracking.

1. Introduction

Sequential Monte Carlo methods, also known as Particle Filters, have demonstrated their success for parameter estimation in nonlinear and non-Gaussian problems in many areas such as video tracking [1], navigation [2], econometrics [3] and signal processing [4]. When the observed data hold the modeling assumptions, the estimated errors converge to zero [5] (i.e. with zero mean and small covariance). However, several sources of error exist that affect the filter performance and lead to inconsistency, where the estimated errors have non-zero mean or high covariance [6]. Filter consistency is commonly analyzed to detect estimation errors over time [7].

Determining the temporal consistency of Particle Filters can be cast as a change detection problem [8]: consistency measures are generated and then analyzed to decide between one of two cases, namely consistent or inconsistent operation. Examples of such approaches include the χ^2 validation [6], the cumulative sum (CUSUM) [9] and the expected model likelihood [10]. Their performance is limited due to drawbacks related to high-dimensional state spaces [6], prior change magnitude assumptions [9] or empirical thresholding [10]. Domain-related knowledge can be exploited to improve change detection performance. For example, in video tracking, Particle Filter consistency is measured as spatial uncertainty [11], time-reversibility [12] or by combining both approaches [13]. However, current approaches are tuned to particular data in the consistency measurement or in the change detection process due to the need of using empirical thresholding approach [12][13].

In order to enable the application of particle filter validation to unseen data, in this paper we propose an approach to estimate its temporal consistency without relying on empirical thresholding and we present a robust model for temporal filter uncertainty. We measure filter consistency as its uncertainty (dispersion of its hypotheses in the state space) and validate an uncertainty model over sliding windows, allowing to increase the robustness of the consistency estimation, unlike existing approaches based on single-point analysis [14][12][13]. Such uncertainty model is approximated by sequential convolutions of mixtures of Gamma distributions whose number of mixture components is selected via modified information-based criteria. By applying hypothesis testing over filter uncertainty models, the parameters required for detecting inconsistency are automatically determined, unlike empirical-based approaches [15][12][13][16]. The proposed approach is included in a framework for online performance evaluation of video tracking [13]. The results show that the proposed approach improves related works over two heterogeneous datasets containing challenges in both change detection and video tracking.

The paper is organized as follows: Section 2 states

☆This work was partially supported by the Spanish Government (EventVideo, TEC2011-25995) and by the EU Crowded Environments monitoring for Activity Understanding and Recognition (CENTAUR, FP7-PEOPLE-2012-IAPP) project under GA number 324359. Most of the work reported in this paper was done at the Centre for Intelligent Sensing in Queen Mary University of London.

*Corresponding author

Email addresses: juancarlos.sanmiguel@uam.es (Juan C. SanMiguel), a.cavallaro@qmul.ac.uk (Andrea Cavallaro)

the addressed problem and Section 3 discusses the related work. Section 4 presents the proposed approach whereas Section 5 describes the filter consistency modeling. Section 6 introduces its use for video tracking evaluation. Section 7 presents and discusses the experimental results. Finally, Section 8 concludes the paper.

2. Problem statement

Let $X_t = \{(x_t^{(n)}, \pi_t^{(n)})\}_{n=1,\ldots,N}$ be the output of N weighted particles generated by a particle filter at time t, where each $x_t^{(n)}$ defines a hypothetical estimation weighted by $\pi_t^{(n)}$. Each particle should have a low (high) weight when it is far from (close to) the ideal state. Each particle is recursively obtained with a prediction, $g(.)$ [17]:

$$x_t^{(n)} = g(x_{t-1}^{(n)}, \kappa_t) \tag{1}$$

and an update step $z(.)$:

$$\pi_t^{(n)} \propto z(x_t^{(n)}, \rho_t), \tag{2}$$

where $\{\kappa_t\}_{t=1\ldots}$ and $\{\rho_t\}_{t=1\ldots}$are independent and identically distributed random processes. From these steps two distributions are derived, namely the prior and posterior distribution. The prior distribution predicts the states (particles) relying on previous data only; whereas the posterior distribution is estimated by considering the prior given all observations up to the current observation time.

The problem we address is the online determination of the consistency of the filter (i.e. its reliability) by observing the posterior distribution. A consistent behavior means that X_t provides an accurate state estimation. Let C and I be the labels for consistency and inconsistency, respectively. The goal of online inconsistency detection is to assign a label l_t as follows:

$$X_t \overset{\varphi(\cdot)}{\rightarrow} l_t \in \{C, I\}, \tag{3}$$

where $\varphi(\cdot)$ is the labeling approach. Such approach should be accurate, offer low latency and operate without manual parameter tuning. Figure 1 shows an example of a particle-filter-based tracker where the filter becomes inconsistent as most of the hypotheses are apart from each other and have small weights. The labeling approach shall automatically identify this inconsistent behavior of the filter.

3. Related work

In this section, we review the literature for estimating the consistency of a Particle Filter and for change detection, which are later particularized for video tracking.

The consistency of a Particle Filter can be estimated from its posterior. For example, likelihood ratios are computed using filter observations (weights) for consistent and inconsistent assumptions, which are later accumulated over time [9]. The Kullback-Leibler divergence is used to measure differences between prior and posterior distributions [10]. However, the prior is assumed to be static without being conditioned to the observed data over time, thus limiting its use to stationary prediction [10], i.e. the variance of the posterior does not increase with time. Filter consistency can be estimated as the dispersion of its hypotheses (particles) in the state space [11]. The posterior hypotheses could be also converted into uniformly distributed variables through the cumulative distribution of the observations [6]. However, its computation for high-dimensional state spaces is not feasible [12]. The Mahalanobis distance (MD) between forward and backward filters can also be used for consistency estimation [12]. However, MD values have not got a fixed variation range for identifying filter inconsistency without ambiguities as several values can simultaneously represent consistent and inconsistent operation under different conditions [13]. Recently, concentration measures have been proposed using the likelihood of the filter observations [19].

Estimating inconsistency of Particle Filters can be approached as a change detection problem. The goal is to recognize significant deviations from a known level of the measurement. Approaches exist based on single or multiple detectors [7]. Single-detector approaches apply a whiteness test to the filter residuals (errors). The cumulative sum approach (CUSUM) is a popular single-detector example that accumulates likelihood ratios of a Particle Filter [9]. Then, empirical thresholding is used to detect changes [9][10][19]. Multi-detector approaches have each detector matched to a certain change assumption. Although not applied to Particle Filters, several approaches exist for signal processing such as the bank of matched filters [7] and Parallel-CUSUM [20]. The former adapts each detector to new change hypothesis when its prediction error is high and the latter runs in parallel several (differently adjusted) CUSUM detectors. Both detect changes by concatenating over time the results of the most probable detectors, namely those with lowest prediction errors. However, CUSUM-based approaches require prior knowledge of the change magnitude, an information that is often not available. For unknown change magnitudes, model validation has been proposed as an alternative when only the unchanged status is known by computing its fitness with observed data [8] such as the χ^2 test to verify uniformity of measurements [6]. Finally, other approaches do not consider any prior modeling or thresholding and use sliding windows for online model extraction and validation. Examples are the two-model validation through the χ^2 test [7] and SVMs [21], where both models are extracted from sliding windows with different lengths, assuming that one does not have any changes. However, online model extraction requires a minimum window length to get statistically significant models.

Particle Filters are widely employed for parameter estimation of targets in video tracking. Filter consistency is analyzed for performance evaluation over time to detect er-

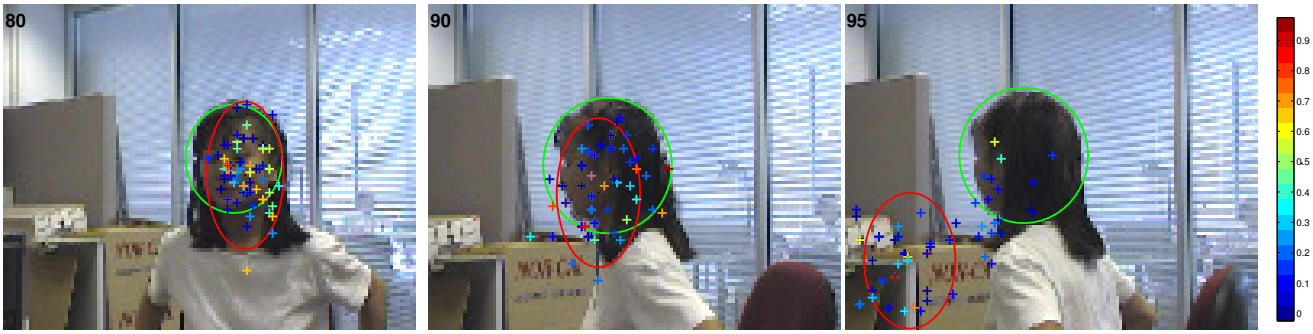


Figure 1: Example of filter consistency for face video tracking using a color-based Particle Filter [18] (with 100 particles). The green ellipse represents the ideal target; the red ellipse represents the estimated target. The left image illustrates a consistent behavior. The central and right image illustrate inconsistent situations. The particles (identified for clarity only by their center) are colored according to their weights: the warmer the color, the higher the weight.

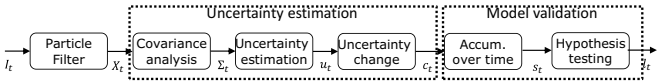


Figure 2: Block diagram of the proposed approach.

rors. Examples are those looking at time-reversibility [12], spatial uncertainty change over time [13], illumination model consistency [22] or filter selection in multi-camera settings [23]. One of the major problems is the use of consistency statistics [11][12] whose variation range is unknown, thus making it difficult to estimate a significant deviation. Empirical thresholding is broadly applied to detect changes, limiting their application to unseen data. Similar approaches also exist for non-Particle Filter-based approaches focused on feature accuracy [14], filter switching [24] or multi-hypothesis similarity [15][16]. All these approaches are making use of data-dependent manually selected thresholds (computed offline) for change detection. This manual tuning prevents the design of online strategies to correct inconsistency thus limiting the analysis of new data.

Table 1 compares the main approaches discussed in this section. Current estimators of Particle Filter consistency are limited as most of the extracted measures do not have a bounded variation range [12], require the knowledge of change magnitudes [9], use empirical thresholding [10][13][22] or are not applicable to large state spaces [6].

4. Accumulated validation of uncertainty

Model validation provides a robust framework for Particle Filter consistency analysis whose performance could be improved by sliding windows. For measuring the consistency of the Particle Filter, we first compute the uncertainty of its posterior and accumulate its change over a temporal window. Then, we validate an uncertainty change model to check consistency (Figure 2). We term the proposed approach as Accumulated Validation of Uncertainty (AVU).

4.1. Filter uncertainty estimation

We estimate the uncertainty for each time t by measuring the spread of the generated hypotheses in the state space through $\Sigma_t = [\zeta_{ij}]$ (the covariance matrix of the filter output X_t), where each element ζ_{ij} is defined as [11]:

$$\zeta_{ij} = \sum_{n=1}^{N} \pi^{(n)} (x_i^{(n)} - \mu_i)(x_j^{(n)} - \mu_j), \tag{4}$$

where $x_i^{(n)}$ is the ith element of the nth estimation (particle) of X_t, $\mu_i = E\left[x_i^{(1\ldots N)}\right]$ and $E[\cdot]$ is the expectation. The filter uncertainty is computed as [13]:

$$u_t = \frac{1}{C} \sqrt[d]{det(\Sigma_t)}, \tag{5}$$

where C is a normalizing constant to consider the target size as in [13], $det(\cdot)$ is the matrix determinant and d is the number of dimensions of $x_t^{(n)}$. Unlike [13], we compute the uncertainty u_t (Eq. 5) for the complete target state without temporal smoothing. Figure 3 shows an example of uncertainty analysis of Particle Filters for video tracking where the filter becomes inconsistent when it loses the target around frame 540.

For detecting uncertainty transitions from low-to-high or high-to-low values, we compute a change signal c_t that maximizes the difference between u_t and previous uncertainty values. We use a sliding window of length W to remove the offset uncertainty value that could be exhibited due to the initial configuration or the observations:

$$c_t = \left| \frac{u_t - u_{\hat{t}}}{u_{\hat{t}}} \right|, \tag{6}$$

where

$$\hat{t} = \underset{j \in W}{argmax} \left(\left| \frac{u_t - u_j}{u_j} \right| \right). \tag{7}$$

4.2. Test statistic and hypothesis testing

The problem consists of detecting changes in the time series $c_t\,(t = 1, 2, \ldots)$, which is sampled from a random

Table 1: Comparison of the reviewed approaches for change detection and model validation. (KEY. CL: Change Length. ET: Empirical Thresholding. GA: Gaussian Assumption. ML: Maximum Likelihood. MV: Model Validation. PF: Particle Filter. VT: Video Tracking.)

Approach	Consistency estimation	Modeling of filter status		Change detection		Usage restrictions	Computationally feasible over time
		No Change	Change	Approach	Sliding win.		
(CUSUM) [7]	Accumulated filter residuals	Offline	Offline	ET	Yes	-	Yes
(χ^2 test) [7]	Gaussian model similarity	Online	Online	MV	Yes	GA	Yes
(Bank filter) [7]	Residuals of Kalman filters	Online	Online	ML	No	CL	Yes
[9]	CUSUM extension for PFs	Offline	Offline	ET	No	PF	Partial
[21]	SVM-based descriptors	Online	No	ET	Yes	-	Yes
[6]	Uniform distribution conversion	Online	No	MV	Yes	PF	Partial
[10]	Expected log likelihood	Offline	No	ET	No	PF	Yes
[20]	Log likelihood ratio	Offline	Partial	ML	No	-	Yes
[12]	Forward-backward similarity	Online	No	ET	No	PF, VT	No
[13]	Spatial uncertainty	Online	No	ET	No	PF, VT	Yes
Proposed	**Filter uncertainty**	**Offline**	**No**	**MV**	**Yes**	**PF**	**Yes**

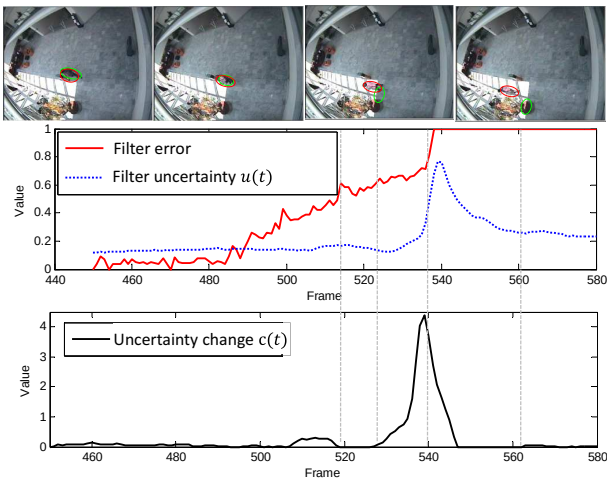


Figure 3: Evolution of the filter uncertainty and its error for color-based Particle Filter video tracking [18]. Green and red ellipses are, respectively, ideal and estimated target locations. Sample frames correspond to vertical dotted lines. The filter error was computed as in Section 7.1.

variable Q following the probability density function (pdf) $p_1(v)$. For increasing robustness of model validation, we accumulate c_t by using a sliding window of length L:

$$s_t = \sum_{r=1}^{L} w_r c_{r+t-L}, \tag{8}$$

where L determines the amount of historical data considered and $w_r \in [0,1]$ weights the contribution of each c_{r+t-L} to s_t defining the amount of variation required for detecting the change. For example, a geometric weight ($w_r = (1-\lambda)\ \lambda^r$ with $\lambda \in [0,1]$) [7] gives low importance to new data, whereas all data are equally considered with a uniform weight ($w_r = 1$). The former case requires higher variation in the new incoming data than the latter for detecting a possible change. We use uniform weights as we are interested in studying changes over time without having any prior assumptions on the filter response (i.e. if changes are long-term or short-term).

Let the null hypothesis H_0 indicate that data are consistent with the model ($s_t \in S$) and s_t be sampled from a random variable S following the pdf $p_2(v)$ which describes the time accumulation of $p_1(v)$, thus defining the test statistic for the H_0 hypothesis. Let H_1 be the hypothesis that a change occurred with unknown magnitude and parameters (i.e. non-trainable). Model validation implies that one of the hypotheses holds true [5]:

$$\begin{array}{l} H_0 : s_t \in S \\ H_1 : s_t \notin S \end{array}, \tag{9}$$

H_1 is accepted (H_0 is rejected) when a change is detected ($s_t \notin S$). For testing the H_0 hypothesis, we use the cumulative distribution function (cdf) of S defined as follows:

$$P_2(j) = \int_{-\infty}^{j} p_2(v)\, dv. \tag{10}$$

For accepting H_0, a probability of false alarms α is required [7] (with values ranging from 0.001 to 0.05 depending on the application) resulting in the following condition:

$$P_2(s_t > \beta) = \alpha, \tag{11}$$

where β is a constant to determine if s_t values follow $p_2(v)$ depending on the considered cdf $P_2(v)$ and the false alarm rate α. Then, Eq. 11 is reformulated as $P_2(|s_t| < \beta) = 1 - \alpha$ to define the hypothesis test for detecting a change in c_t as the condition $s_t > \beta$. The value of β can be (approximately) determined by computing the empirical distribution of s_t and estimating the α-quantile.

In summary, the proposed approach relies on estimating the cdf $P_2(v)$, which depends on the pdfs $p_1(v)$ and $p_2(v)$ of the data c_t and s_t, respectively, and the computation of β to accept ($l_t = C$) or reject ($l_t = I$) H_0.

5. Modeling the consistency of the filter

We now model the consistent filter status $p_1(v)$ and its window-based accumulation $p_2(v)$.

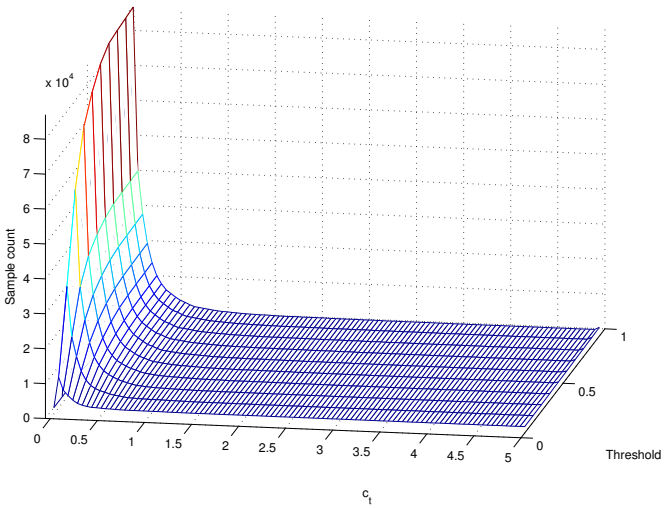


Figure 4: Histogram of c_t values with filter error $e(x_t^E, x_t^{GT}) < \tau$ (H_0 hypothesis) for 100 runs of a color-based Particle Filter tracker [18] over the dataset from [13]. The filter error $e(.) \in [0, 1]$ is defined as Eq. 18, between the ideal x_t^{GT} and the estimated x_t^E states; τ is a threshold to define the consistent case (H_0 hypothesis).

5.1. Filter uncertainty model $p_1(v)$

Obtaining the PF uncertainty change c_t considers three stages. In the first stage, the weighted covariance matrix of the target state is computed as described in Eq. 4. Each matrix element is a weighted sum of products between two terms $(x_i^{(n)} - \mu_i)$ and $(x_j^{(n)} - \mu_j)$. Each term can be modeled as a Random Variable (RV) following a zero-mean Normal distribution $\mathcal{N}(0, \sigma_i)$ as defined in Eq. 1. The variance σ_i of each distribution depends on the process noise κ_t as defined in Eq. 1. Therefore, Eq. 4 is a weighted combination of products between two Gaussian-distributed RVs that can be expressed as a combination of Chi-Square RVs (or their equivalent form using Gamma RVs) [25]. Such weighted covariance matrix only depends on the common representation of the posterior density in the PF framework (set of particles and associated weights), thus being applicable to any PF-based tracker. The second and third stages (Eqs. 5 and 6, respectively) consist of pairwise subtractions, products and ratios of Gamma RVs. The result of each operation can be expressed via Bessel functions which are mixtures of Gamma distributions [26][27].

We propose to model $p_1(v)$ as a mixture of K Gamma distributions [28]. In this mixture, each kth Gamma is defined by its parameters (η_k and θ_k) and its contribution (γ_k) to the mixture ($\sum_{k=1}^{K} \gamma_k = 1$). The pdf $p_1(v)$ is then approximated by:

$$gm(v; \xi) = \sum_{k=1}^{K} \gamma_k f(v; \eta_k, \theta_k), \tag{12}$$

where v is a data sample, $\xi = \{\langle \eta_k, \theta_k, \gamma_k \rangle : k = 1, ..., K\}$ and $f(v; \eta_k, \theta_k)$ is the kth Gamma distribution defined as:

$$f(v; \eta, \theta) = \frac{1}{\theta^\eta} \frac{1}{\Gamma(\eta)} v^{\eta-1} e^{-\frac{v}{\theta}} \quad \text{for } v \geq 0 \text{ and } \eta, \theta > 0 \tag{13}$$

where η is the shape and θ is the scale. When $\eta > 1$ the distribution is bell-shaped, whereas for $\eta < 1$, it is L-shaped. The parameter set ξ is estimated using training data. Figure 4 depicts examples of $p_1(v)$ distribution using empirical data for various assumptions of the H_0 hypothesis or consistent case (i.e. filter error). The c_t values are always positive with an L-shaped distribution close to zero (the uncertainty is almost constant for the consistent case) for all considered cases. To compute the mixture parameters ξ, we used an expectation-maximization (EM) approach based on maximizing the log-likelihood of the hypothesized models [29]. However, EM approach does not correctly determine the optimum number of K components as the likelihood can always be increased by adding more components to the mixture [28]. Standard goodness-of-fit tests (e.g., χ^2 and Kolmogorov-Smirnov [30]) only consider the likelihood without penalizing the number of parameters and therefore they are not valid for optimum mixture modeling.

For choosing K, we modify the Akaike and Bayesian Information criteria [28] (AIC and BIC, respectively) that penalize models with a high number of parameters. AIC minimizes the Kullback-Leibler distance between the true data distribution and the hypothesized distribution. BIC extends AIC by considering the number of data samples. Both criteria include two terms: one depends on the log-likelihood and the other penalizes models with more parameters. However, their variation range is different. The likelihood term depends on the number of samples[1] ($L_K = \sum_{i=1}^{n} gm(v_i; \xi)$) whereas the penalization term has none (for AIC) or low (for BIC) dependency on the number of samples. Hence, the final decision is completely driven by the log-likelihood as penalization costs do not influence enough for large number of samples. Thus, selecting the hypothesized model with highest likelihood.

In order to make equal the influence of the two terms in the final selection, we introduce a variable penalization cost to produce the modified AIC, $dAIC$:

$$\hat{K}_{dAIC}(d) = \underset{K}{argmin} \left(-2ln\left(L_K\right) + d \cdot 2v_K\right), \tag{14}$$

where d represents the weight of the penalization cost and, for each K-Gamma mixture, v_K is its number of parameters and L_K is its maximized likelihood. The modified BIC, $dBIC$ is defined as follows:

$$\hat{K}_{dBIC}(d) = \underset{K}{argmin} \left(-2ln\left(L_K\right) + d \cdot v_K ln(n)\right), \tag{15}$$

where n is the sample size. We do not change the likelihood term, $-2ln\left(L_K\right)$, as it is the deviance, a measure

[1] The EM results are independent of the sample size as the maximization is performed over the likelihood variations among the hypothesized models.

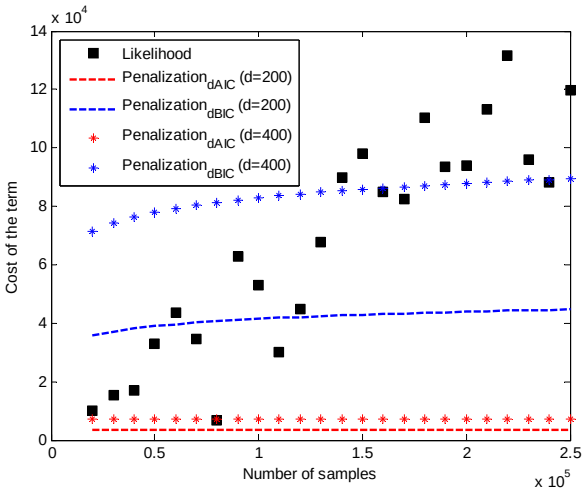


Figure 5: Comparison of likelihood and penalization terms for proposed K selection criteria ($dAIC$ and $dBIC$) using $d = \{200, 400\}$. Results were computed for a mixture of two Gammas.

of lack of fit for a model [28]. Then, the optimum K is selected as the most frequent K when scanning the results obtained for different values of $d = 1, ..., D$. Note that $d = 1$ corresponds to the original AIC and BIC. Moreover, at a certain d value, the variable penalization cost is higher than that of the likelihood and the first model ($K = 1$) is always selected. Hence, these wrong selections should not be considered for K optimum selection. Figure 5 illustrates an example of the modified criteria showing that likelihood and penalization terms are comparable.

5.2. Accumulated filter uncertainty model $p_2(v)$

In this paper we propose the use of convolution to approximate the upper bound of the cumulative distribution of s_t [31]. Applying the proposed hypothesis test in Section 4 uses the cumulative distribution of s_t ($P_2(v)$, see Eq. 11), which requires to estimate the joint distribution of the accumulated and dependent RVs, $p_2(v)$. This distribution is unknown and its estimation is not straightforward to be analytically solved. Therefore, approximations of this distribution are needed to employ the proposed approach.

One approximation could be to empirically generate this $p_2(v)$ using real data as done for the distribution $p_1(v)$ of the statistic c_t. However, this option is limited in two aspects. First, a large number of samples is required to provide an accurate distribution estimation, thus requiring a large training set which cannot be always guaranteed. Second, each window length needs a different distribution $p_2(v)$ (see Eq. 8), thus increasing the complexity of the training process. For example, a PF tracker to be analyzed with window lengths between 1 and 50 frames requires to get 50 different $p_2(v)$.

On the other hand, we do not need a precise shape estimation for the true distribution of $p_2(v)$ as we are only interested in the rightmost part of $P_2(v)$ to perform the hypothesis testing. Hence, approximations of the upper bounds for the sum of the c_t statistic are more suitable for the proposed approach. We formulate the accumulation of L c_t values as the L-sum of Q_i RVs where all Q_i have the same distribution $p_1(v)$. Such upper bound can be estimated for finite ($E[Q_i] < \infty$), dependent, non-negative and real-valued RVs [31]: these conditions satisfied by all Q_i. Hence, the distance between the sum of dependent RVs and the sum of their independent duplicates (i.e. assuming independent Q_i) is upper-bounded by a certain factor which depends on the correlation between Q_i's (their dependence) and their mean values [31]. We exploit this conclusion to use convolution as an approximation of the upper bound of $P_2(v)$, which allows to estimate the cut value for the hypothesis test (β parameter in Eq. 11). The use of convolution allows to quickly estimate the cut value for any desired length of the sliding window only requiring the $p_1(v)$ distribution.

After assuming independent Q_i to compute such upper bound, we use the convolution approach [32] to get the pdf of the sum of two random variables, with pdfs $m_1(v)$ and $m_2(v)$, as their convolution $m_3 = m_1 * m_2$ given by:

$$m_3(v) = \sum_{j=-\infty}^{\infty} m_1(j) \cdot m_2(v - j), \tag{16}$$

for $v = ..., -2, -1, 0, 1, 2, ...$ Hence, we can exploit such property to compute $p_2(v)$ as a L-fold convolution of $p_1(v)$:

$$p_2(v) = p_1^{(L)}(v) = p_1^{(L-1)}(v) * p_1(v), \tag{17}$$

where $p_1^{(0)}(v) = p_1(v)$ and $p_1(v) * p_1^{(0)}(v) = p_1(v)$. Although this recursive convolution can be analytically approached, the existing proposals are based on combinatorial analysis [33] that heavily increases the computational cost of the convolution. According to this, we decide to obtain this convolution by empirically estimating $p_1(v)$ (i.e. generating random samples of $p_1(v)$ for computing its pdf) and then, performing the standard L-fold convolution as described in Eq. 17.

We do not consider the central limit theorem (CLT) [32] to approximate $p_2(v)$ as it depends on the number of added random variables to estimate $p_2(v)$ as $\mathcal{N}(L\mu, \sqrt{L}\sigma^2)$ where $\mu = E[Q]$ and $\sigma^2 = var(Q)$. The proposed approach may use short windows where CLT accuracy decreases. Figure 6 shows some examples of the empirical cdf $P_2(v)$ and their approximations assuming independence (convolution and CLT). Although both approximations do not reflect the empirical cdf (consequently the pdf $p_2(v)$), they allow to establish an upper bound for $P_2(v)$. Moreover, CLT is outperformed by convolution for short window lengths, thus decreasing the accuracy of the upper bound estimation.

6. Accumulated performance evaluation of video tracking

We combine AVU into an online method performance evaluation of Particle Filter-based video tracking, ARTE [13].

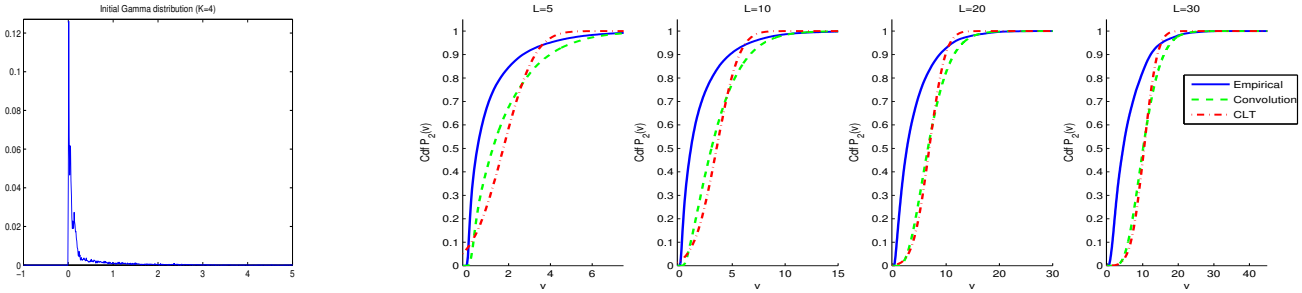


Figure 6: Approximated cumulative distribution function (cdf) $P_2(v)$ (right) for window lengths $L = 5-30$ using a mixture of $K = 4$ gammas (left). For each plot, data corresponds to the obtained $P_2(v)$ via the estimated $p_2(v)$ with the Empirical, Convolution and Central Limit Theorem (CLT) approaches using the data described in Sec. 7.1.

Table 2: Description of the proposed modification on ARTE.

Approach	**# Thresholds**	**Acquisition**	**Signals analyzed**
ARTE [13]	τ_1, τ_2, τ_3	Manual	$C_t^{W_1,\hat{t}}$, $C_t^{W_2,\hat{t}}$, $C_t^{W_1,t}$, $C_t^{W_2,t}$
ARTE*	β_1, β_2	Automatic	c_t

ARTE determines whether the Particle Filter is successfully estimating the target state without the use of ground-truth. ARTE analyzes the Particle Filter consistency and the time-reversibility property of target motion.

Similarly to Eq. 6, ARTE defines four change signals ($C_t^{W_1,\hat{t}}$, $C_t^{W_2,\hat{t}}$, $C_t^{W_1,t}$ and $C_t^{W_2,t}$) based on spatial uncertainty (i.e. only considering the center of the target location) that combines different window lengths and considerations of the change reference (first or last sample in the window). In particular, it monitors slow (W_1) or sudden (W_2) increases ($\hat{t}$) or decreases (t) of the uncertainty. Then, change detection is applied over these four signals by empirical thresholding to detect when the Particle Filter posterior is inconsistent. First, the threshold τ_1 is applied to $C_t^{W_1,\hat{t}}$and $C_t^{W_2,\hat{t}}$, for positive changes (consistent-inconsistent). Negative changes (inconsistent-consistent) are detected by using the threshold τ_2 on $C_t^{W_1,t}$ and $C_t^{W_2,t}$. Then, a third one τ_3 is applied to $C_t^{W_2,t}$ for negative small changes that indicate increases of the Particle Filter consistency (i.e. its posterior is becoming more accurate). For simplifying such tuning, the thresholds were originally defined based on $\tau_1 : \tau_2 = -\tau_1$ and $\tau_3 = {}^{-\tau_1}/_2$.

The proposed modification (hereinafter called ARTE*) aims to substitute ARTE's change detectors with our proposal. We calculate s_t and detect consistency by uncertainty-based validation over sliding windows, which defines a threshold β_1 computed as in Sec. 4.2. Low-to-high (positive) and high-to-low (negative) transitions are detected as, respectively, consistent-to-inconsistent and inconsistent-to-consistent changes of s_t. For small negative changes, we include another validation for detecting inconsistent-to-consistent changes of c_t with a threshold $\beta_2 = \beta_1/2$. Table 2 summarizes the modification showing that fewer signal detections are required and the thresholds are automatically computed.

7. Experimental results

In this section we first compare the results of the proposed AVU and related approaches for analyzing the consistency of particle filters and then, we evaluate the use of AVU for online performance evaluation of video tracking[2].

7.1. Experimental setup

Let us consider a color-based Particle Filter for video tracking [18] with $x_t^{(n)}$ being a five-component vector composed of the target position, the two main axes and the orientation of the bounding ellipse approximating its area on the image plane. Color histograms are used as target model and are generated in the RGB space for pedestrian (P) and car (C) targets and in the HSV one for face targets (F), using 8x8x8 bins in both cases. The filter parameters are $N = 400$ (particles) and the variances for target center $\sigma_{x,y} = 5$, size $\sigma_{H_x,H_y} = 0.75$, orientation $\sigma_\theta = 4°$ and appearance noise $\sigma_c = 0.2$. For AVU, we consider the false alarm rate $\alpha = 0.005$ to accept the H_0 hypothesis.

We use two evaluation sets (D1 and D2) with sequences selected from the following datasets: CAVIAR[3], PETS2001[4], PETS2010[5], CLEMSON[6], VISOR[7], AVSS2007[8], TRECVID[9] and MIT TRAFFIC[10]. D1 is the same set as in [13], which is composed of 18 sequences ($\sim$ 3400 annotated frames). D2 contains 51 sequences ($\sim$ 7500 annotated frames). Both datasets include three target types, namely cars, people and faces, and present challenging situations for tracking such as total or partial occlusions, clutter, and illumination or scale changes. The characteristics of the two sets are summarized in Table 3. Sample frames (and target initialization) for D2 are shown in Figure 7 (for D1, we use the same initialization as shown in [13]).

[2]Additional results, video sequences and software implementations can be found at http://www-vpu.eps.uam.es/publications/PFConsistency.

[3]http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

[4]http://www.cvg.rdg.ac.uk/PETS2001/

[5]http://www.cvg.rdg.ac.uk/PETS2010/

[6]http://www.ces.clemson.edu/~stb/research/facetracker

[7]http://imagelab.ing.unimore.it/visor/

[8]http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html

[9]http://www.itl.nist.gov/iad/mig/tests/trecvid/2011/

[10]http://www.ee.cuhk.edu.hk/~xgwang/MITtraffic.html

Table 3: Summary of the evaluation sets and their tracking challenges (Key. SC: Scale changes. AC: Appearance changes. IC: Illumination changes. O:occlusions. C:clutter.)

Set	**Dataset**	**Target**	**Size**	**Tracking challenges**
D1	CAVIAR	P1 – P4	384x288	IC, C
	PETS2001	P5 – P10	768x576	SC, O, C
	PETS2010	P12 – P18	768x576	O, C
	CLEMSOM	F1 – F4	128x196	SC, AC, C, O
	VISOR	F5, F6	352x288	SC, C, O
D2	AVSS2007	P1 – P4	720x576	O, C, SC
	CAVIAR	P5 – P6	384x288	SC, C
	PETS2010	P7 – P19	768x576	SC, IC, O
	TRECVID	P20 – P24	720x576	IC, O, C
	VISOR	F1 – F4	352x288	IC, O, C
	TRECVID	F5 – F10	720x576	IC, O
	MIT CAR	C1 – C16	720x480	AC, IC, O, C

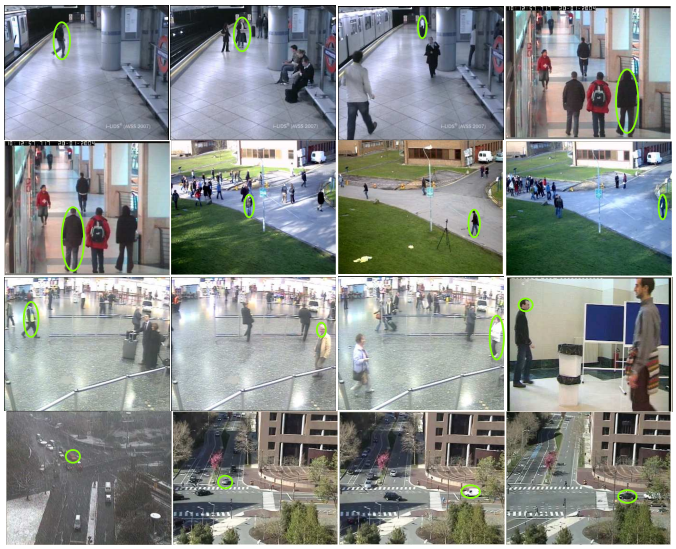


Figure 7: Sample target initialization for the evaluation set D2. From top-left to bottom-right: Pedestrian targets: *AB_Easy_man* (P1), *AB_Hard_man* (P2), *AB_Medium_woman* (P3), *ThreePastShop2cor* (P5), *ThreePastShop2cor* (P6), *S2_L1_v01* (P7), *S2_L2_v01* (P8), *S2_L3_v01* (P11) and *Trecvid* (P20-P21); face targets: *Trecvid* (F1), *Visor_occ_1* (F2); car targets: *Dtneu_redcar* (C1), *Mv2_020_silcar* (C3), *Mv2_020_whtvan* (C6) and *Mv2_020_blkcar* (C12).

7.2. Evaluation measures

To analyze the accuracy for detecting uncertainty changes, we define the ground-truth changes δ_t as the time instants when the filter error, $e_t \in [0,1]$, changes from successful ($e_t < 1$) to unsuccessful ($e_t = 1$) or viceversa. For video tracking, we define e_t as the spatial tracking error [11]:

$$e_t(x_t^E, x_t^{GT}) = 1 - \frac{2\left|A_t^E \cap A_t^{GT}\right|}{\left|A_t^E\right| + \left|A_t^{GT}\right|}, \tag{18}$$

where x_t^E and x_t^{GT} are the estimated and ideal target locations at time t; $\left|A_t^E \cap A_t^{GT}\right|$ is their spatial overlap (in pixels); and $\left|A_t^E\right|$ and $\left|A_t^{GT}\right|$ represent their area (in pixels). For obtaining δ_t, we first identify when the filter is inconsistent by binarizing $e_t(x_t^E, x_t^{GT})$ as follows:

$$e_t^b = \begin{cases} 1 & if \quad e_t(x_t^E, x_t^{GT}) = 1 \\ 0 & if \quad e_t(x_t^E, x_t^{GT}) < 1 \end{cases}. \tag{19}$$

Then, we assume a consistent start of the filter ($\delta_0 = 0$) and obtain each δ_t as the initial and ending instants of the inconsistency operation:

$$\delta_t = \left|e_t^b - e_{t-1}^b\right|, \forall t > 0. \tag{20}$$

Let TP and FP be the generated changes that match (TP) or not (FP) with ground-truth ones δ_t for each time t. A match is allowed within a tolerance window of ± 5 frames. Let FN be the unmatched δ_t. To evaluate detection performance, we compute Precision (P), Recall (R) and F-score (F):

$$P = TP/(TP + FP), \tag{21}$$

$$R = TP/(TP + FN), \tag{22}$$

$$F = 2 \cdot P \cdot R/(P + R). \tag{23}$$

To evaluate the performance of online tracking evaluation, we focus on the temporal segmentation task (i.e. determining whether the tracker is successful) by means of the Receiver Operating Characteristic (ROC) analysis. ROC analysis requires the definition of an ideal (manual) segmentation to compute the similarity between the generated and the ideal segmentation. A successful track is determined when the error $e_t(x_t^E, x_t^{GT})$, defined as in Eq. 18, is $e_t < 1$. An unsuccessful track is identified by $e_t = 1$.

7.3. Uncertainty modeling

We use the data of the color-based PF tracker [18] to obtain the pdf $p_1(v)$ of Q, which is then convolved to get $p_2(v)$. Different subsets of c_t values are employed to estimate the pdfs which are extracted from:

$$\{X_1, ..., X_T | e_t(x_t^E, x_t^{GT}) < \tau\}, \tag{24}$$

where $X_{1...T}$ are T filter posteriors and τ is a threshold that defines the consistent case.

We first compare the pdfs generated by D1 and D2. Figure 8 depicts the similarity between the pdfs $p_2(v)$ obtained for D1 and D2 using the KS statistic [30]. Multiple $p_2(v)$ are considered depending on the allowed ground-truth error of c_t values (via the threshold τ). For $L = 1$, $p_2(v)$ corresponds to $p_1(v)$. Low τ values have the lowest similarity (i.e. highest KS values). The tracker rarely has low ground-truth errors and, therefore, a reduced number of samples is used to estimate $p_2(v)$, which decreases its accuracy. High τ values slightly increase the dissimilarity as c_t values of PF inconsistency are included to model $p_2(v)$ (i.e. before the tracker loses the target). Mid-range τ values get the highest similarity which have a balance between the number of samples and inconsistency of c_t values. Finally, the two-sample KS test [30] determined

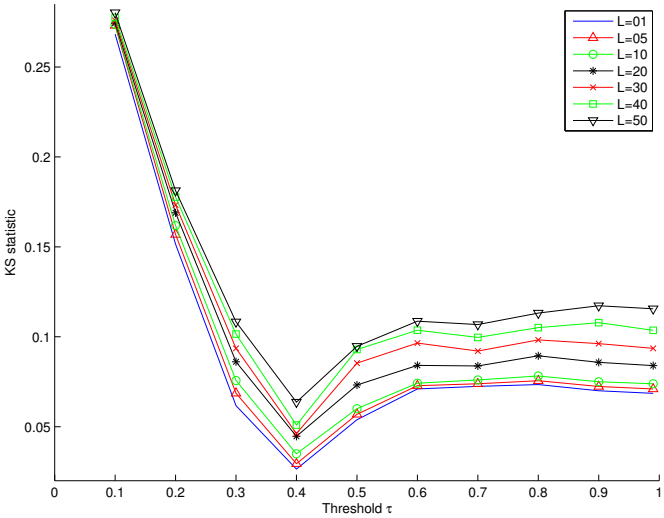


Figure 8: Similarity between the $p_2(v)$ pdf estimated for D1 and D2 datasets. Data employed considers various window lengths L and uses s_t values with different filter errors (τ). The Kolmogorov-Smirnov statistic ($KS \in [0, 1]$) is employed as similarity measure.

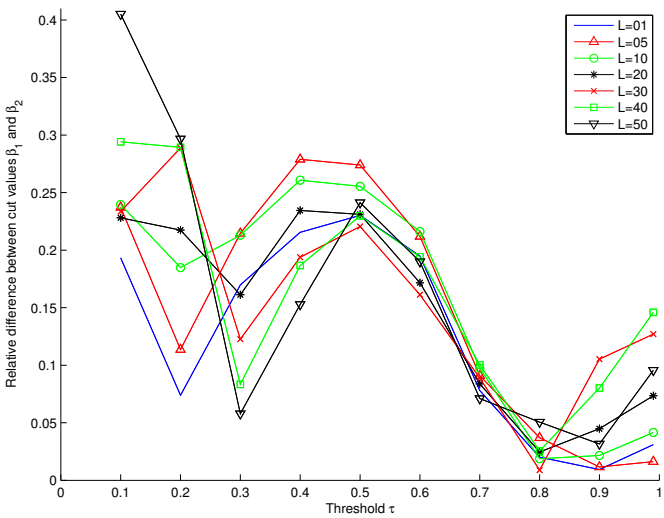
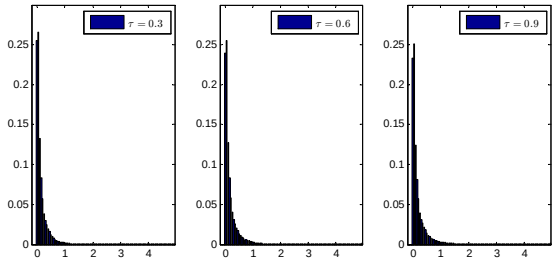


Figure 9: Relative difference between the cut values β estimated for D1 and D2 datasets. Data employed considers various window lengths L and uses s_t values with different filter errors (τ).

that the $p_2(v)$ pdfs for D1 and D2 are different distributions. This suggests that empirical thresholding may not efficiently detect changes for both datasets simultaneously.

Nevertheless, we are interested in the upper bound of the $P_2(v)$ cdf (see Section 4.2) instead of an accurate $p_2(v)$ estimation. Figure 9 shows the difference between the cut values β obtained for D1 and D2 (β_1 and β_2, respectively). We consider the difference $D = (\beta_1 - \beta_2)/min(\beta_1, \beta_2)$ as a similarity measure between the results. We observe that high (low) τ values provide the highest (lowest) similarity as more (less) samples are considered to estimate $P_2(v)$. These results suggest that $p_1(v)$ should be modeled using



Distribution	Kolmogorov-Smirnov statistic		
	$\tau = 0.3$	$\tau = 0.6$	$\tau = 0.9$
Exponential	.164	.188	.207
Gamma	**.092**	**.077**	**.075**
Log-normal	.234	.195	.190
Poisson	.831	.807	.784
Rayleigh	.556	.586	.623

Figure 10: Distribution fitting for c_t values with different filter errors (τ) using the Kolmogorov-Smirnov test. Bold are best results.

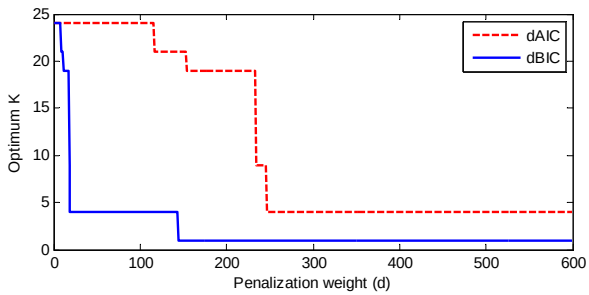


Figure 11: Selected optimum K for each weight of the penalization term (for models with $K = 1...25$).

c_t values with an associated error between $\tau \in [0.7, 0.9]$.

We compute filter uncertainty data for the consistent status, with a window length $W = 25$, by running the filter 100 times over D1. Then, we extract the c_t values corresponding to H_0 (1938032 examples in total) that are represented in Figure 4 for each τ value (computed as indicated in Eq. 18). Figure 10 shows the pdfs for the extracted data with $\tau = \{0.3, 0.6, 0.9\}$ and the fitting results for well-known distributions using the two-sample Kolmogorov-Smirnov (KS) test [30] where Gamma is the best one in all cases. After that, we use the data for $\tau = 0.9$ (highest error) for uncertainty modeling. Although its KS values are the lowest ones, the significance level of the KS test indicates that the Gamma fitting is not perfectly accurate and therefore, motivating the proposed modeling with a mixture of K Gamma distributions [28].

To select the optimum K for the mixture, we use $dAIC$ and $dBIC$ (Eqs. 14 and 15, respectively). Figure 11 shows the results for weights d ranging from 1 to 600: $dBIC$ has higher penalization costs than $dAIC$ when evaluating models with high K, quickly converging to $K = 1$. However, if we exclude $K = 1$ as explained in section 5.1, both criteria agree on the optimum $K = 4$ for modeling $p_1(v)$.

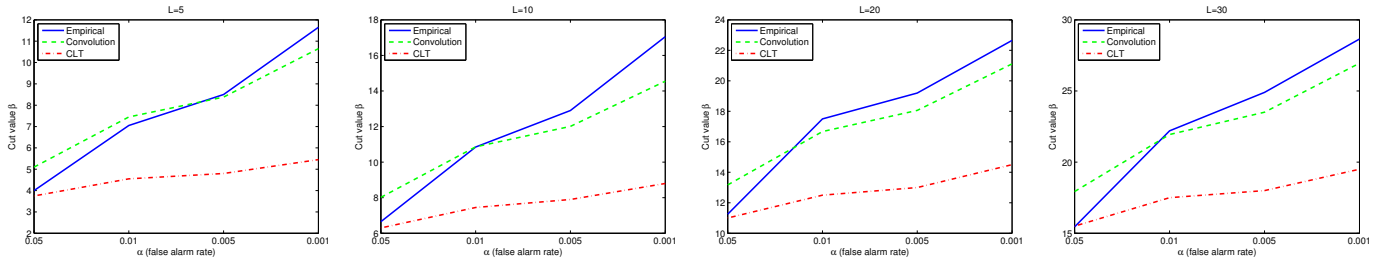


Figure 12: Comparison of selected cut values β for hypothesis testing using the approximations of the $p_2(v)$ pdf based on the empirical, convolution and CLT approaches (all based on the $p_1(v)$ learned with D1 dataset).

Figure 12 compares the results of selected cut values β for the hypothesis testing under different false alarm rates (α) and using the approaches to compute $p_2(v)$ described in Sec. 5.2. The empirical results represent the optimum β values to be approximated. These values are better estimated by the convolution approach for the various false alarms of the H_0 hypothesis. The total accumulated difference in results between empirical and convolution approaches is 34.75 whereas between empirical and CLT is 86.15, thus demonstrating the preferred used of the convolution. Moreover, the error of both approaches increases when decreasing the false alarm rate α which shows the limitations of the upper-bound approximation.

7.4. Change detection results

We compare AVU against representative approaches for online change detection without thresholding: the χ^2 two-model sliding window ($Two-MChi$) [7] that assumes Gaussian-distributed data, the bank of Kalman filters adapted to various change hypothesis ($Mmodel$) [7] and the empirical thresholding approach ($EmpTh$) [13], which is tuned using D1. All approaches are applied to the uncertainty change signal c_t obtained as described in section 4.1. Experiments with different lengths of the sliding window (L) are performed for testing the robustness of AVU and the results are summarized in Figure 13 for the D1 and D2.

Results for D1 are shown on the left part of Figure 13. In general, F-score results demonstrate that AVU outperforms the selected state-of-the-art approaches for any length L showing stable F-scores around 0.40 (with a performance peak for $L = 5$ and $L = 10$). $M-model$ is able to detect several δ_t (high recall) as it generates many changes in the uncertainty signal (low precision). $Two-MChi$ get best results for $L = 5$ and $L = 10$ but heavily decreases its performance for large L values because of the unsatisfaction of the data Gaussianity assumption. Compared to optimum thresholding ($EmpTh$), it can be observed that for most of the values of $L = \{1, 20, 30, 40, 50, 60\}$, AVU detects less δ_t (having lower recall). However, AVU clearly outperforms $EmpTh$ as its precision is higher, resulting in a high F-score compared to $EmpTh$. Moreover, AVU presents a slight decrease in F-score as L increases. Although larger L values increases AVU's precision, its recall decreases as a higher amount of change is required in the sliding window. Additionally, Particle Filter uncertainty is not usually high for long periods of time as for video tracking, Particle Filters tend to estimate the state of the most similar object to the target after inconsistency (thus, becoming consistent). Hence, large L values do not improve the overall performance.

The results for D2 (right column of Figure 13) present similar conclusions to the ones for D1 where AVU also improves the selected approaches for any window lengths. However, AVU's results show a different pattern as for D1. Unlike D1, the performance peak (considering F-score) is not centered around $L = \{5, 10\}$ being shifted towards $L = \{20, 30\}$. This can be explained because of two reasons. The first one is that the filter data seems to be very stable (47 ground-truth changes in average for each run which contains 51 targets, ~ 0.92 errors/target) compared to D1 (~ 1.21 errors/target) indicating that D2 is easier to analyze than D1. Hence, filter errors in D2 are more significant helping the change detection task. The second reason regards the duration of the change, D2 sequences are longer and the filter rarely finds similar objects in the image after becoming inconsistent (thus, not changing to the consistent status). On the other hand, the filter estimation changes from consistent to inconsistent (and viceversa) for some targets of D1 (more frequently than in D2), thus making more difficult the change detection task.

Figure 14 shows an example of the compared approaches applied over the uncertainty signal (black) with the objective of detecting the ground-truth changes (cyan). On the left column, a ground-truth change is defined ($\delta_{89} = 1$) which is correctly detected by all the approaches. However, $M-model$ generates additional detections for every small change in the uncertainty signal. Observe that although $EmpTh$ and AVU correctly detect the change, the length is shorter for $EmpTh$ as it does not use any sliding windows. The right column describes a situation when the filter shows a small inconsistency (frames 25-40) that does not produce a ground-truth change. $EmpTh$, $Two-MChi$ and $M-model$ wrongly detect such change as an uncertainty variation whereas AVU does not due to the use of sliding windows allowing to tolerate a certain amount of change before detecting it.

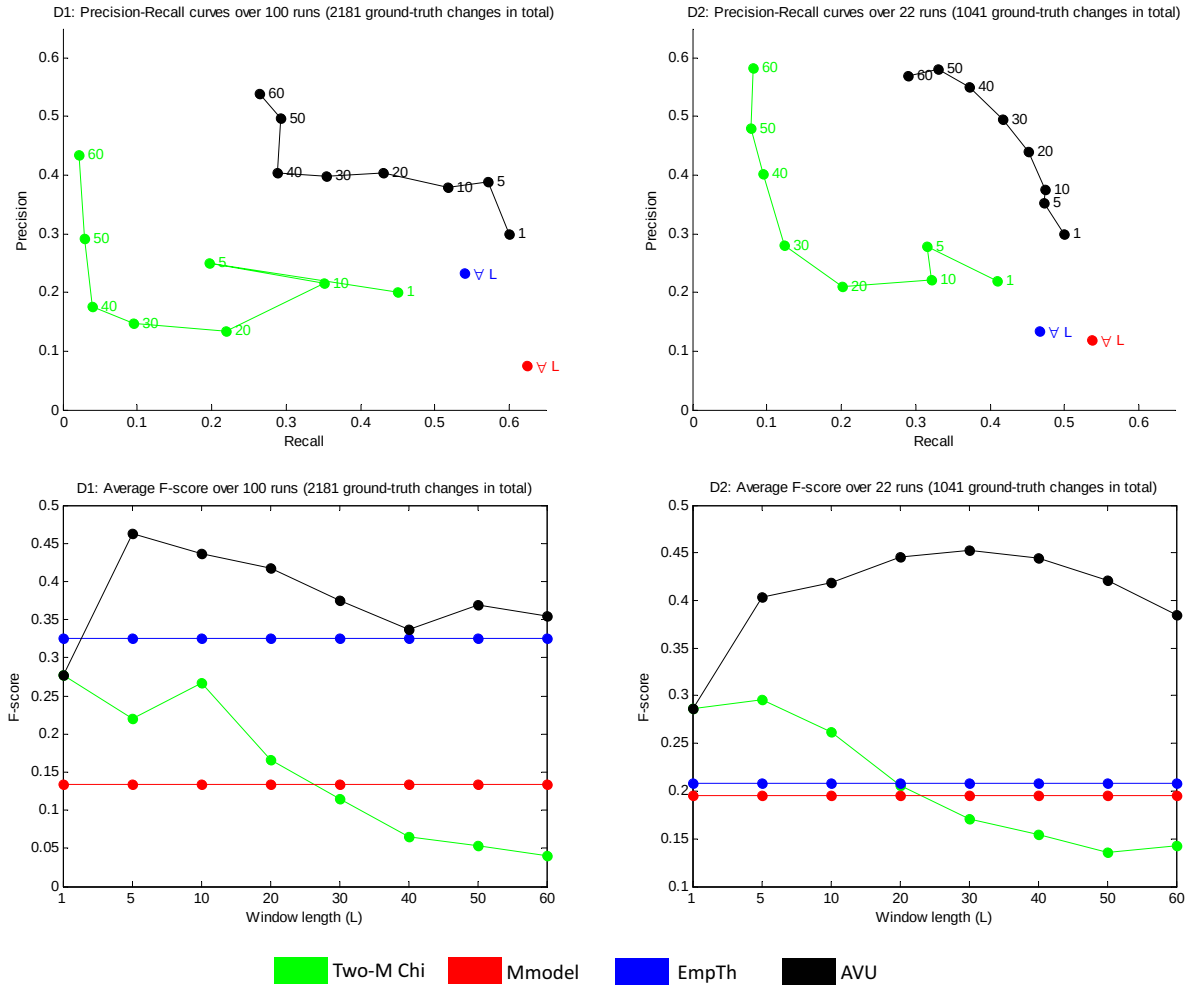


Figure 13: Comparison for selected change detection approaches with different lengths (L) of the sliding windows for evaluation sets D1 (left) and D2 (right).

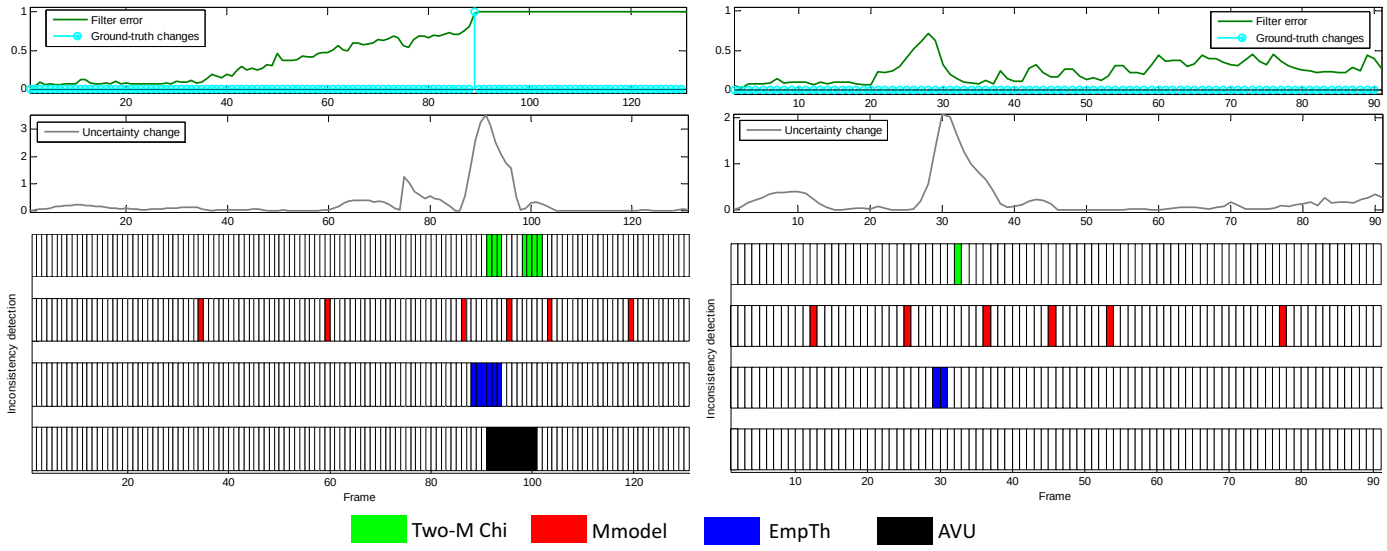


Figure 14: Example of change detection for determining Particle Filter consistency for targets P1 (left) and P6 (right).

Table 4: ROC analysis for successful-unsuccessful segmentation of video tracking for sets D1 (left) and D2 (right). Data are presented as mean ± standard deviation. (Key. ARTE: Adaptive Reverse Tracking Evaluation [13]; ARTE*: threshold-automatic ARTE; AUC: area under the curve, FPR: false positive rate, TPR: true positive rate).

Approach	AUC	TPR	FPR	Approach	AUC	TPR	FPR
ARTE [13]	$.772 \pm .06$	$.717 \pm .05$	$.172 \pm .02$	ARTE [13]	$.747 \pm .06$	$.732 \pm .14$	$.237 \pm .05$
ARTE* (L=5)	$.770 \pm .07$	$.737 \pm .05$	$.197 \pm .03$	ARTE* (L=5)	$.770 \pm .07$	$.800 \pm .14$	$.261 \pm .04$
ARTE* (L=10)	$.785 \pm .05$	$.799 \pm .03$	$.228 \pm .01$	ARTE* (L=10)	$.806 \pm .05$	$.900 \pm .10$	$.289 \pm .03$
ARTE* (L=20)	$.766 \pm .04$	$.779 \pm .04$	$.247 \pm .02$	ARTE* (L=20)	$.763 \pm .04$	$.926 \pm .08$	$.401 \pm .03$
ARTE* (L=30)	$.750 \pm .07$	$.760 \pm .05$	$.235 \pm .01$	ARTE* (L=30)	$.735 \pm .07$	$.868 \pm .14$	$.398 \pm .03$
ARTE* (L=40)	$.743 \pm .06$	$.739 \pm .03$	$.224 \pm .01$	ARTE* (L=40)	$.717 \pm .06$	$.796 \pm .12$	$.361 \pm .03$
ARTE* (L=50)	$.700 \pm .02$	$.765 \pm .03$	$.236 \pm .03$	ARTE* (L=50)	$.722 \pm .06$	$.767 \pm .14$	$.279 \pm .05$
ARTE* (L=60)	$.698 \pm .04$	$.723 \pm .06$	$.300 \pm .04$	ARTE* (L=60)	$.710 \pm .06$	$.776 \pm .14$	$.276 \pm .04$

Table 5: Comparison of execution times for temporal segmentation with ARTE and ARTE* using 10 runs for datasets D1 and D2. Data are presented as mean ± standard deviation.

Approach	Execution time per frame (ms)		
	Min	Max	Mean
ARTE [13]	2.2 ± 0.25	397.6 ± 110.04	4.58 ± 1.25
ARTE* (L=5)	2.4 ± 0.33	409.7 ± 111.32	3.9 ± 0.75

7.5. Track quality estimation

The results of the method described in Sec. 6 for online evaluation are presented in Tables 4 and 5.

The left part of Table 4 shows that ARTE* has similar accuracy to ARTE for D1. A noticeable improvement in TPR is observed for ARTE* with all lengths. However, ARTE* slightly increases the False Positive Rate compared to ARTE because of the use of the sliding window, requiring a higher amount of variation to detect an uncertainty change. This implies in some situations a short delay in the detection of changes. ARTE* reaches similar performance to that of the change detector of ARTE whose threshold values were *manually tuned on the same dataset* (D1). The right part of Table 4 (results on D2) shows a situation where the thresholds of ARTE are not optimum. As it can be observed, shorter windows got higher results than that of ARTE demonstrating that the proposed approach generalizes better than the optimum thresholding of ARTE. However, a performance decrease is observed as the length of the window increases due to the reduction of the number of detected changes. The main advantage of ARTE* over ARTE is that it does not require to set any thresholds.

In Table 5, we can observe the effect of the proposed approach in the computational cost of the track quality estimator. The most noticeable difference is the reduction of the mean processing time around 15%, from 4.58 ms (ARTE) to 3.9 ms (ARTE*). As ARTE* detects a smaller number of (false) changes than ARTE, it avoids the analysis of the stages for checking the origin of such changes, i.e. the tracker has failed, recovered after a failure or focused on a distractor object.

Figure 15 illustrates a comparative example for online evaluation of video tracking. A car target is tracked

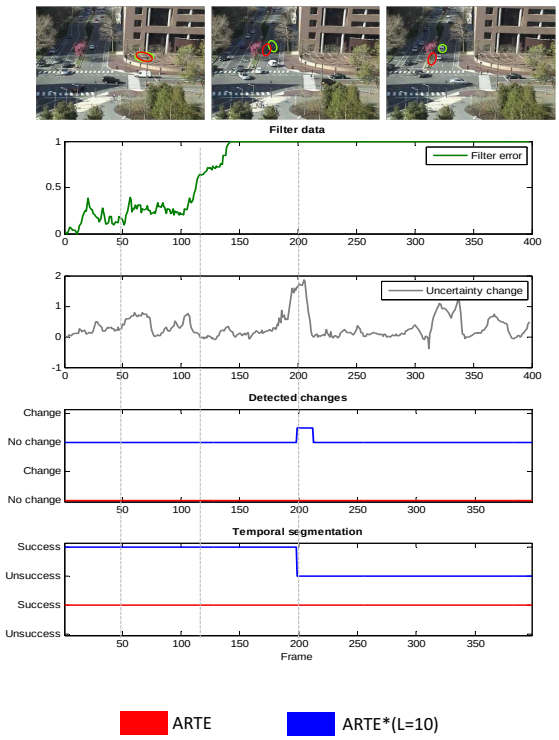


Figure 15: Online evaluation example for a color-based particle filter tracker to detect successful and unsuccessful results using ARTE [13] and the proposed approach ARTE*. Images correspond to frames 50, 120 and 200 of *Mv2_020_whitevan* (target C16). Green ellipse: ideal target location; red ellipse: estimated target location. The filter error is computed as described in Section 7.1.

throughout the sequence and as the filter error indicates, the Particle Filter loses the target around frame 150 due to scale changes and similar objects. At this frame, the uncertainty signal change is not noticeable and therefore, no changes are detected. Then, a gradual change appears in the uncertainty around frame 200 due to a shadow. Only ARTE* is able to detect it and correctly perform a good segmentation of filter success.

7.6. Application to other trackers

We demonstrate the generality of the proposed approach by evaluating two state-of-the-art trackers [34][35].

Table 6: Comparison of change detection approaches for the selected PF-based trackers. Best results are indicated in bold. Data are presented as mean ± standard deviation. (Key. P: Precision, R: Recall, F: F-score).

(a) Dataset D1

Approach	Color-tracker [18]			Frag-tracker [34]			Sparse-tracker[35]		
	P	R	F	P	R	F	P	R	F
Two-M Chi	.133 ± .01	.220 ± .02	.166 ± .01	.186 ± .03	.251 ± .02	.214 ± .01	.080 ± .01	.788 ± .04	.145 ± .03
Mmodel	.074 ± .03	**.624 ± .01**	.133 ± .02	.134 ± .02	.541 ± .01	.215 ± .01	.104 ± .02	.718 ± .03	.182 ± .02
EmpTh	.233 ± .06	.539 ± .03	.326 ± .04	.142 ± .01	.530 ± .03	.224 ± .02	.102 ± .06	.410 ± .04	.163 ± .04
AVU (L=20)	**.404 ± .03**	.430 ± .04	**.417 ± .02**	**.264 ± .04**	**.587 ± .02**	**.364 ± .03**	**.264 ± .09**	**.503 ± .03**	**.346 ± .05**

(b) Dataset D2

Approach	Color-tracker [18]			Frag-tracker [34]			Sparse-tracker[35]		
	P	R	F	P	R	F	P	R	F
Two-M Chi	.210 ± .00	.202 ± .00	.206 ± .00	.139 ± .00	.071 ± .00	.094 ± .00	.080 ± .00	.525 ± .00	.138 ± .00
Mmodel	.119 ± .00	**.537 ± .00**	.195 ± .00	.119 ± .00	.245 ± .00	.160 ± .00	.099 ± .00	.475 ± .00	.164 ± .00
EmpTh	.134 ± .00	.466 ± .00	.208 ± .00	.134 ± .00	.245 ± .00	.173 ± .00	.102 ± .00	.468 ± .00	.167 ± .00
AVU (L=20)	**.440 ± .00**	.451 ± .00	**.446 ± .00**	**.328 ± .00**	**.263 ± .00**	**.292 ± .00**	**.253 ± .00**	**.728 ± .00**	**.375 ± .00**

Table 7: Comparison of video tracking performance evaluation for the selected PF-based trackers. Data are presented as mean ± standard deviation. (Key. ARTE: Adaptive Reverse Tracking Evaluation [13]; ARTE*: threshold-automatic ARTE; AUC: area under the curve, FPR: false positive rate, TPR: true positive rate).

Tracking approach	Dataset D1						Dataset D2					
	ARTE [13]			ARTE* (L=20)			ARTE [13]			ARTE* (L=20)		
	AUC	TPR	FPR	AUC	TPR	FPR	AUC	TPR	FPR	AUC	TPR	FPR
Color-tracker [18]	.772 ± .06	.717 ± .05	.172 ± .02	.766 ± .04	.779 ± .04	.247 ± .02	.747 ± .06	.732 ± .14	.237 ± .05	.763 ± .04	.926 ± .08	.401 ± .03
Frag-tracker [34]	.727 ± .07	.748 ± .13	.294 ± .04	.746 ± .07	.738 ± .09	.246 ± .06	.715 ± .02	.822 ± .04	.390 ± .02	.788 ± .04	.718 ± .06	.143 ± .04
Sparse-tracker [35]	.723 ± .05	.684 ± .05	.239 ± .13	.742 ± .06	.798 ± .05	.315 ± .09	.720 ± .06	.812 ± .04	.371 ± .13	.775 ± .06	.730 ± .05	.180 ± .11

The first tracker models targets as fragments adaptively selected over time which are embedded in the PF framework [34]. The second tracker performs multi-hypothesis estimation based on sparse appearance models, presenting a PF-like structure [35]. We employ the code provided by the authors with the default parameter settings. For the proposed approach, we learn $p_1(v)$ for each tracker using D1 dataset and we use $L = 20$ as a compromise between the previously described results for $D1$ and $D2$ datasets. The $EmpTh$ approach is tuned to get best results for D1. The presented results are the mean of 10 runs.

Table 6 summarizes the results of uncertainty change detection for the selected trackers. AVU gets the highest Precision and Recall scores for all trackers in most of the cases as compared to the selected change detection approaches. The Precision increase of AVU is due to the use of the sliding window to filter noise and the modeling of the uncertainty signal. The results also exhibit low Precision values for all trackers, indicating that the uncertainty signal is difficult to analyze and many false positives are generated. AVU's Recall is also improved in many sequences as slow changes are also considered within the window. Recent trackers [34][35] often employ mechanisms to gradually adapt the target model over time and, therefore, the uncertainty slowly changes as tracking failures are integrated in the target model. Analysis over sliding windows improves performance proportionally to the adaptation rate (high for [35] and low for [34]) as observed in the corresponding results.

Table 7 compares the results for track quality estimation. The proposed approach has two effects for online performance evaluation of tracking. First, it reduces the FPR due to the improved accuracy for the detected changes in filter consistency. This effect can be observed for ST in D2 dataset and for FT in D1 and D2 datasets. Second, it also improves TPR as less false changes are generated and therefore the online evaluator has to analyze less (possibly) wrong changes which may lead to evaluation errors. This effect is observed for ST in D1 dataset and for PF in D1 and D2 datasets.

8. Conclusions

We presented an online estimation of particle filter consistency that uses a sliding-window-based hypothesis testing approach and models filter uncertainty as convolutions of mixtures of Gamma distributions. Compared to manual thresholding, the proposed approach increased the precision and maintained the recall values. We applied the proposed approach to online evaluation of video tracking, without the need of ground truth data. Experiments show that the proposed approach generalizes better than the corresponding threshold-based solution. Results also indicate that filter inconsistency does not last long time in video tracking, which requires to use short window lengths. The high precision values of the proposed approach allows us to reduce the overall computational time as a smaller number of detections are generated. Finally, the results over recent video trackers demonstrate the flexibility of the proposed approach.

Although our approach was demonstrated on particle filter, it can be applied to other multi-hypothesis filters

that allow the measurement of the spread of its hypotheses (i.e, representing its posterior target estimation as set of samples and associated weights). As future work, we will explore its application to deterministic filters through appropriate adaptations [15][36], model validation based on multiple detectors and the selection of the optimum window length for a particular particle filter setting.

References

[1] E. Maggio, A. Cavallaro, Video tracking: theory and practice, Wiley, 2011.

[2] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P.-J. Nordlund, Particle filters for positioning, navigation, and tracking, IEEE Trans. on Signal Process. 50 (2) (2002) 425–437.

[3] J. Geweke, Bayesian inference in econometrics models using monte carlo integration, Econometrica 33 (1) (1989) 1317–1339.

[4] A. Doucet, X. Wang, Monte carlo methods for signal processing, IEEE Signal Process. Mag. 22 (6) (2005) 152–170.

[5] Y. Bar-Shalom, X. R. Li, T. Kirubarajan, Estimation with applications to tracking and navigation, Wiley, 2001.

[6] F. Van der Heijen, Consistency checks for particle filters, IEEE Trans. Pattern Anal. Mach. Intell. 1 (1) (2006) 140–145.

[7] F. Gustafsson, Adaptive Filtering and Change Detection, Wiley, 2000.

[8] C. Andrieu, A. Doucet, S. Singh, V. Tadic, Particle methods for change detection, system identification, and control, Proc. IEEE 92 (3) (2004) 423–438.

[9] B. Azimi, P. Krishnaprasa, A particle filtering approach to change detection for nonlinear systems, EURASIP J. Adv. Signal Process. 2004 (15) (2004) 2295–2305.

[10] N. Vaswani, Additive change detection in nonlinear systems with unknown change parameters, IEEE Trans. on Signal Process. 55 (3) (2007) 859 –872.

[11] E. Maggio, F. Smerladi, A. Cavallaro, Adaptive multifeature tracking in a particle filtering framework, IEEE Trans. Circuits Syst. Video Technol. 17 (10) (2007) 1348–1359.

[12] H. Wu, A. Sankaranarayanan, R. Chellappa, Online empirical evaluation of tracking algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 32 (8) (2010) 1443–1458.

[13] J. C. SanMiguel, A. Cavallaro, J. M. Martínez, Adaptive online performance evaluation of video trackers, IEEE Trans. Image Process. 21 (5) (2012) 2812 –2823.

[14] C. Erdem, Sankur, A. Tekalp, Performance measures for video object segmentation and tracking, IEEE Trans. Image Process. 13 (7) (2004) 937–951.

[15] V. Badrinarayanan, P. Perez, F. Le Clerc, L. Oisel, On uncertainties, random features and object tracking, in: Proc. IEEE Int. Conf. Image Process., San Antonio (TX, USA), 2007, pp. 61–64.

[16] T. Biresaw, C. Regazzoni, A bayesian network for online evaluation of sparse features based multitarget tracking, in: IEEE Int. Conf. on Image Process., Orlando (USA), 2012.

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

[18] K. Nummiaro, E. Koller-Meier, E. Van Gool, An adaptive colour-based particle filter, Image Vis. Comput. 2 (1) (2003) 99–110.

[19] C. Maiz, E. Molanes-Lopez, J. Miguez, P. Djuric, A particle filtering scheme for processing time series corrupted by outliers, IEEE Trans. Signal Process. 60 (9) (2012) 4611–4627.

[20] C. Li, H. Dai, H. Li, Adaptive quickest change detection with unknown parameter, in: IEEE Int. Conf. on Acoust., Speech and Signal Process., 2009, pp. 3241–3244.

[21] F. Desobry, M. Davy, C. Doncarli, An online kernel change detection algorithm, IEEE Trans. Signal Process. 53 (8) (2005) 2961–2974.

[22] S. Das, A. Kale, N. Vaswani, Particle filter with a mode tracker for visual tracking across illumination changes, IEEE Trans. Image Process. 21 (4) (2012) 2340–2346.

[23] Y. Zhou, H. Nicolas, J. Benois-Pineau, A multi-resolution particle filter tracking in a multi-camera environment, in: IEEE Int. Conf. on Image Process., Cairo (Egypt), 2009, pp. 4065–4068.

[24] P. Pan, F. Porikli, D. Schonfeld, Recurrent tracking using multifold consistency, in: Proc. of IEEE Int. Work. on Perf. Eval. of Tracking and Surv., Miami (USA), 2009.

[25] R. Ware, F. Lad, Approximating the distribution for sums of products of normal variables, University of Canterbury, England, Tech. Rep. UCDMS 15 (2003) 2003.

[26] M. K. Simon, Probability distributions involving Gaussian Random Variables, Springer, 2002.

[27] C. S. Withers, S. Nadarajah, On the product of gamma random variables, Quality & Quantity 47 (1) (2013) 545–552.

[28] G. McLachlan, D. Peel, Finite mixture models, Vol. 299, Wiley-Interscience, 2000.

[29] T. Benaglia, D. Chauveau, D. R. Hunter, D. Young, mixtools: An R package for analyzing finite mixture models, J. Stat. Softw. 32 (6) (2009) 1–29.

[30] G. Corder, D. Foreman, Nonparametric statistics for non-statisticians: a step-by-step approach, Wiley, 2009.

[31] C. A. Charalambides, M. Koutras, N. Balakrishnan., Probability and statistical models with applications, Chapman& Hall/CRC, 2001.

[32] C. Grinstead, J. Snell, Introduction to probability, Amer Mathematical Society, 1997.

[33] J. Sarabia, F. Prieto, C. Trueba, The n-fold convolution of a finite mixture of densities, Appl. Math. Comput. 218 (19) (2012) 9992 – 9996.

[34] E. Erdem, S. Dubuisson, I. Bloch, Fragments based tracking with adaptive cue integration, Computer Vision and Image Understanding 116 (7) (2012) 827–841.

[35] D.Wang, H. Lu, M.-H. Yang, Online object tracking with sparse prototypes, IEEE Trans. Image Process. 22 (1) (2013) 314–325.

[36] J. C. SanMiguel, A. Cavallaro, J. M. Martínez, Standalone evaluation of deterministic video tracking, in: IEEE Int. Conf. on Image Process., Orlando (USA), 2012.