To appear in IEEE Transactions on Image Processing

Adaptive on-line performance evaluation of video trackers

Juan C. SanMiguel, Andrea Cavallaro and José M. Martínez

***Abstract*—We propose an adaptive framework to estimate the quality of video tracking algorithms without ground-truth data. The framework is divided into two main stages, namely the estimation of the tracker condition to identify temporal segments during which a target is lost and the measurement of the quality of the estimated track when the tracker is successful. A key novelty of the proposed framework is the capability of evaluating video trackers with multiple failures and recoveries over long sequences. Successful tracking is identified by analyzing the uncertainty of the tracker, whereas track recovery from errors is determined based on the time-reversibility constraint. The proposed approach is demonstrated on a particle filter tracker over a heterogeneous dataset. Experimental results show the effectiveness and robustness of the proposed framework that improves state-of-art approaches in the presence of tracking challenges such as occlusions, illumination changes and clutter, and on sequences containing multiple tracking errors and recoveries.**

***Index Terms*—video tracking, track quality, tracking uncertainty, time-reversibility, failure detection, particle filter.**

I. Introduction

VIDEO tracking is an important step in many applications, such as video surveillance, human-computer interaction, traffic monitoring, video indexing and object-based video compression. The video data to be analyzed present high complexity and variability because of pose changes, illumination variations, occlusions and clutter. Under such conditions, no single video tracker can perform perfectly in all situations and failures are expected in real tracking scenarios. An online track failure detector and quality estimator is therefore needed to measure tracking performance over time.

Common tracking performance evaluations use *empirical discrepancy methods* [1] that compare off-line ground-truth data with the estimated target state. Ground-truth data are expensive to produce and therefore usually cover only short temporal segments of test video sequences, thus representing only a small percentage of data variability. This limitation makes it difficult to extrapolate performance evaluation results to (unlabeled) new sequences. Moreover, evaluation using ground-truth is unfeasible for on-line performance analysis [2]. To extend the applicability of performance evaluation, *empirical standalone methods* (ESM) for track-quality estimation without ground-truth data have been defined for large unlabeled datasets, self-tuning (automatic control via on-line analysis), comparative ranking and tracker fusion. ESMs are based on properties of the estimated trajectories, such as motion smoothness [3], area consistency [4], or time-reversibility [5]; statistical properties of the tracker output, such as observation likelihood [6], spatial uncertainty [7], or consistency checks [8]; complementary features, such as color contrast [9] or background discriminative power [10]; and combinations of these properties [11]. However, these approaches are generally application-dependent [3, 4, 9, 10], not applicable to long sequences [5] or non-adaptive to errors and recoveries of the tracker [6, 7]. Therefore their use and experimental validation is generally limited to short or low-complexity videos.

To overcome the above-mentioned limitations, we propose a novel adaptive empirical standalone method for track-quality estimation that is applicable to image sequences with multiple tracking errors and recoveries. The proposed framework is based on a two-stage adaptive strategy that first determines when a target is being successfully tracked (temporal segmentation) and then estimates track quality during successful tracking. The framework effectively combines the filter uncertainty and the time-reversibility constraint of a tracker to measure the quality of the estimated target state. The analysis of the filter uncertainty allows one detecting unstable tracking results and the detection of a recovery after a tracking failure by applying a reverse tracker. We demonstrate the proposed approach in a particle filter framework over a heterogeneous dataset with sequences containing tracking challenges such as occlusions, clutter and appearance changes. A block diagram of the proposed framework is shown in Fig. 1.

This paper is organized as follows. Section II discusses related works. Section III describes the identification of the target condition, whilst Sec. IV introduces how we estimate track quality. Section V discusses the results and comparisons with alternative approaches. Finally, Sec. VI summarizes the paper.

II. Prior work

Empirical standalone methods for the evaluation of track quality can be classified into three main categories, namely trajectory-based, feature-based and hybrid [12].

This work was partially supported by the Spanish Government (TEC2007-65400 SemanticVideo), Cátedra Infoglobal-UAM for "Nuevas Tecnologías de video aplicadas a la seguridad", Consejería de Educación of the Comunidad de Madrid and European Social Fund. Most of the work reported in this paper was done during two research stays of the first author at Queen Mary University of London under a research grant of Universidad Autonóma de Madrid.
J.C. SanMiguel & J.M. Martínez are with the TEC Department, Universidad Autonóma de Madrid ({juancarlos.sanmiguel,josem.martinez}@uam.es). A. Cavallaro is with the School of Electronic Engineering and Computer Science, Queen Mary University of London (andrea.cavallaro@eecs.qmul.ac.uk).

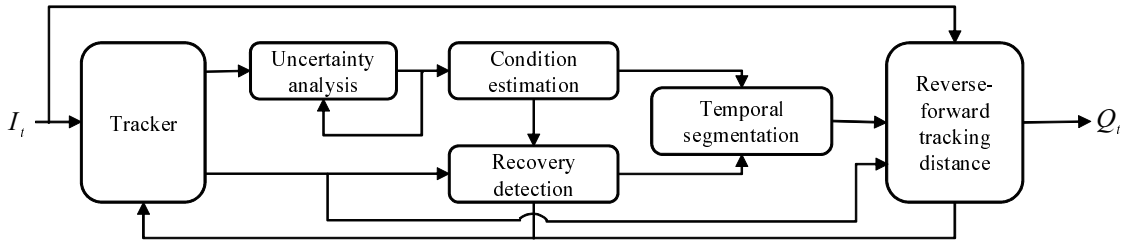


Fig. 1. Block diagram of the proposed adaptive on-line performance evaluation approach for video trackers.

Trajectory-based measures use information from the estimated trajectories to quantify the quality of a tracker and can in turn be grouped into three sub-categories: model-based, forward, and reverse measures. *Model-based measures* (MM) rely on on-line learning of trajectory models. Track quality is computed as the similarity between models and new object trajectories [13, 14]. MMs need a considerable amount of data to learn the models thus limiting their applicability for evaluation. *Forward-based measures* (FM) threshold features extracted from the estimated trajectory in short time windows. Examples of features are trajectory length [4] and smoothness of target velocity [4, 15, 16] or of direction of change [3, 15, 17]. FMs generally provide only a binary decision and are application-dependent, thus limiting their field of applicability. *Reverse measures* (RM) rely on the time-reversibility constraint of the motion of physical objects. A tracking analysis in reverse time direction is applied to measure track quality with different strategies, such as on a frame-by-frame basis using template matching [18] or on the full trajectory length using the Kanade-Lucas-Tomasi tracker [11] or a particle filter [5]. This idea can be extended by reflecting the two tracker analyses to the specific time instant to be evaluated [19]. Although RMs have been found to be preferable to other trajectory-based approaches [12], their applicability is limited to short sequences as they suffer from error accumulation (short-length version) or are computationally unfeasible (full-length version). The determination of the optimal reverse analysis temporal window will provide a solution for the analysis of long sequences, as we propose in this paper.

Feature-based measures analyze internal stages or the output of a tracker and quantify feature difference or feature consistency. Methods based on *feature differences* (FD) estimate track quality by considering feature variations related to background/foreground color differences [10, 20] or boundary contrast along the target contour [9, 21]. However, as this variation cannot be guaranteed in all types of scenarios (e.g. when targets are similar to the background), FDs are application dependent and inadequate for assessing general-purpose trackers. Methods based on *feature consistency* (FC) compute statistics to validate feature values over time and may look at shape [22], scale [15] or appearance consistency [3, 9, 23, 24, 25]. Furthermore, probabilistic trackers provide an estimation of the target state that is exploited to compute statistics related to the observation likelihood [6, 26, 27], the covariance of the target state [6, 7, 28, 29] or statistical tests (Chi-Square [8] and Kolmogorov-Smirnov [30]). FCs based on probabilistic tracking using the target state representation outperform other approaches [12]. However, they fail when the target moves across areas with varying levels of clutter that affect the observation likelihood. Moreover, when the tracker follows distractors (objects with similar features to those of the target) it might maintain the same level of observation likelihood. A mechanism to determine these tracking conditions on-line is therefore important for an evaluation method to work adaptively.

Finally, *hybrid* measures combine previously described approaches. Smoothness in both direction and motion can be combined with color consistency [3]. Likewise, an equally weighed combination of time-reversibility evaluation and feature difference using color histogram and sum of square differences error estimation, respectively, can be used [11, 31]. Finally, multiple measures such as motion smoothness, trajectory complexity, shape and color consistency can be used to produce multiple track quality estimations [4, 15].

A summary of the track quality evaluation categories described in this section is given in Table I.

III. Is the tracker on target?

A fundamental yet very challenging task is to determine when tracking is successful: one needs to establish whether a tracker is *correctly* estimating the target state at each time instant or it is estimating the state of another physical process corresponding to a portion of the image that is not representing the target. In the former case the tracker is *on-target*, whereas in the latter the tracker is *on-background*. We differentiate two cases of tracker-on-background, namely when the tracker is estimating the state of a distractor (an object with similar features to those of the target) and when the tracker is recovering from a failure.

A. Problem modeling

To describe the tracker condition, we define three events, here referred to as locked-on, locking-in and scanning. The *locked-on* event describes the tracker while *following* an object, which can be the target or a distractor. The *locking-in* event refers to the tracker adapting its estimation to an object after a failure or when the track is better adjusted to (closing-in) the target. Finally, the *scanning* event describes the tracker searching an object after a tracking failure has happened.

We determine the tracker condition using a modeling based on finite-state machines (FSM). A FSM is represented by a directed graph $\mathcal{G} = \langle \mathcal{S}, \mathcal{E} \rangle$ where $\mathcal{S}$ is the set of nodes representing the states and $\mathcal{E}$ is the set of transitions from one

Table I
TRACK QUALITY ESTIMATORS (KEY. D: DETERMINISTIC. P:PROBABILISTIC)

Category	Sub-category	Features	Measures	Trackers	References
Trajectory	Forward	Size & position	Euclidean	D & P	[3][4] [15-17]
	Model	Position	Euclidean	D & P	[13][14]
	Reverse	Position & state-space model	Mahalanobis & Euclidean	D & P	[5][11][18][19]
Feature	Difference	Position & contour	Bhattacharyya & Euclidean	D & P	[3][9][10]
	Consistency	Size, appearance & state-space model	Inf. Theory & change detection	P	[6-8][15][18][20-30]
Hybrid	-	Size, position & appearance	Euclidean	D & P	[3][4][11][15][30]

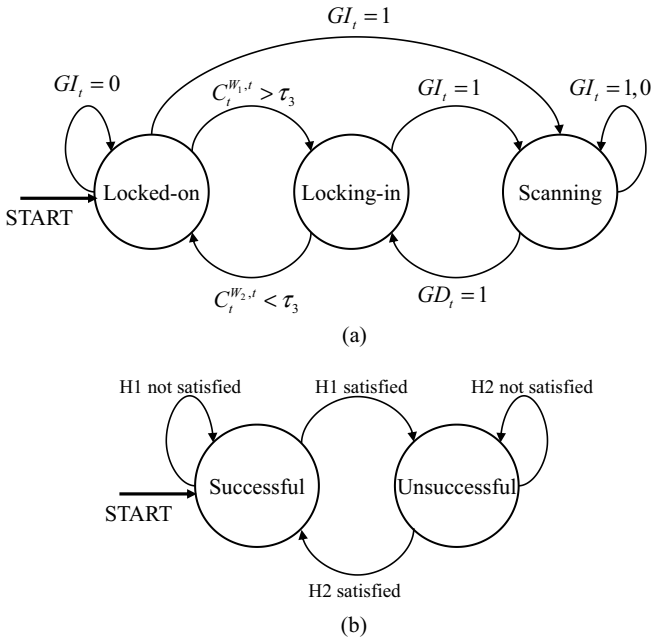


Fig. 2. (a) The finite-state machine used to determine the condition of a tracker. The conditions are: *locked-on*, when the algorithm is tracking the target or a distractor (an object similar to the target), *scanning*, when the algorithm is searching the target after a failure, and *locking-in*, when the algorithm is re-focusing on the target or a distractor during a recovery. (b) The finite-state machine used to determine the temporal segmentation in *successful* and *unsuccessful* tracking.

state to another. The state diagram of the tracker-condition FSM is depicted in Fig. 2(a).

Next, based on the established tracker condition, we segment time windows of operation of a tracker based on whether the algorithm is *on-target* (successful) or on *on-background* (unsuccessful). This temporal segmentation is modeled with a second FSM whose state diagram is depicted in Fig. 2(b). The transitions between the states of the two FSMs are defined as described in the following sections.

B. Uncertainty analysis

Let the target state, x_t, at time t, be defined as

$$x_t = f(I_t, x_{t-1}, \beta_{t-1}), \tag{1}$$

where $f(.)$ represents the tracking algorithm, I_t the video frame at time t, β_{t-1} the model of the target[1] to track at time $t-1$ and x_{t-1} the target state estimation at time $t-1$.

Based on its widespread use in video tracking, let us consider Bayesian filtering as example of a tracker. In particular, we will use a framework defined for elliptical color-based particle filtering [32]. The state x_t is a vector whose elements define the position, the two axes and the orientation of an ellipse on the image plane, whereas the model β_{t-1} is a color histogram. The output of the filter at each time step is the sample set $X_t = \left\{(\mathbf{x}_t^{(n)}, \pi_t^{(n)})\right\}_{n=1,\dots,N}$ of N weighted particles, where each particle $\mathbf{x}_t^{(n)}$ represents one hypothetical state of the target that is weighted by $\pi_t^{(n)}$, according to the similarity of its features to those of the model [32].

The uncertainty of the tracker is used as indicator of unstable periods of the output data (e.g. wrong target estimation) providing information about the conditions discussed in Sec. III-A. We measure the tracker uncertainty using the spatial uncertainty of the N particles (i.e. the spread of the particles). This uncertainty is estimated by analyzing the eigenvalues of the covariance matrix $C_t = [c_{ij}]$ [7], where for simplicity of notation we omit the time index t from each element of the matrix. The elements of the matrix are defined as

$$c_{ij} = \sum_{n=1}^{N} \pi^{(n)} \mathrm{E}\left[(x_i^{(n)} - \mu_i)(x_j^{(n)} - \mu_j)\right], \tag{2}$$

where $\pi^{(n)}$ is the weight of each particle n; $x_i^{(n)}$ ($x_j^{(n)}$) is the i^{th} (j^{th}) element of the n^{th} particle; N is the number of particles; $i, j = 1, ..., d$; and d is the number of dimensions of the state vector. In the specific case mentioned above, the state is composed of five elements and therefore we compute a 5×5 covariance matrix. Consequently, the spatial uncertainty, S_t, is defined as [7]:

$$S_t = \sqrt[d]{det(C_t)}, \tag{3}$$

where $det(\centerdot)$ represents the determinant of a matrix. Note that if the state contains additional elements such as target dynamics (e.g. velocity), the covariance matrix has to be computed considering only those elements related to the spatial location.

The tracker uncertainty $\tilde{U}_t$ is finally obtained by normalizing the spatial uncertainty using width, H_x, and height, H_y, of the target (i.e. the axes of the ellipse):

$$\tilde{U}_t = \frac{S_t}{4H_xH_y}. \tag{4}$$

Note that this uncertainty measure is independent of target size and of number of samples. A temporal filtering stage is

[1]If the model of the target does not change after initialization, then β_{t-1} is replaced with β_{t_0}

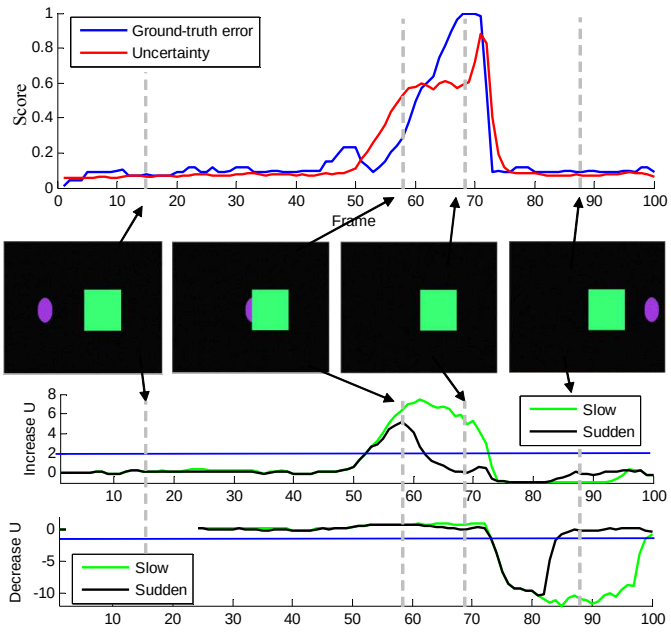


Fig. 3. Evolution of tracking filter uncertainty and ground-truth error for a toy sequence. Blue lines in the bottom plots indicate the value of the threshold applied ($\tau_1 = 2$ for increases and $\tau_2 = -2$ for decreases). The ground-truth error signal was computed as described in Sec. V-B.

finally applied to smooth the result:

$$U_t = \alpha U_{t-1} + (1-\alpha)\tilde{U}_t. \tag{5}$$

where $\alpha \in [0,1]$ determines the update rate of the uncertainty signal. Low values of α produce a fast update.

The tracker is expected to maintain a constant or slightly decreasing value of uncertainty (indicating a temporal refinement of target estimation) when it is successfully tracking the target. The uncertainty of the filter increases when the tracker loses the target. Finally, a decrease of the filter uncertainty after a tracking failure indicates that the tracker has locked on an object, which might be the correct target or a distractor.

An example of use of the uncertainty is depicted in the toy example of Fig. 3. The color-based particle filter tracks a magenta solid ellipse that moves from left to right and is occluded for a few frames by a square (Figure 3, middle). It can be observed that the time window during which the uncertainty increases and decreases reflects what one could estimate using ground-truth data (Figure 3, top).

C. Tracker condition estimation

We aim to detect temporal changes in uncertainty levels in order to discriminate transitions of the tracker condition between locked and not locked, at each time instant. In fact, small uncertainty levels indicate that the tracker is locked on an object (i.e. the particles are concentrated in an area close to this object) that might be the target or a distractor, whereas large uncertainty levels indicate that the tracker is scanning the image while searching for an appropriate candidate object to lock on (i.e. the particles are spread over large areas).

To detect *uncertainty level transitions* while removing the offset value that could be exhibited by the tracking algorithm, we define a change signal, C_t^W, which maximizes the difference between filter uncertainty at time t, U_t, and previous uncertainty values within a time window W:

$$C_t^{W,k} = \frac{U_t - U_{\hat{t}}}{U_k}, \tag{6}$$

where

$$\hat{t} = \underset{j \in W}{argmax} \left(\left| \frac{U_t - U_j}{U_k} \right| \right) \tag{7}$$

and $k \in \{\hat{t}, t\}$, with $k = \hat{t}$ for detecting low-to-high *uncertainty level transitions* and $k = t$ for detecting high-to-low *uncertainty level transitions*. The size of the window W determines the speed of response of the operator. Large windows allow detecting slow changes but introduce a delay in the filter response when the signal recovers from a no-change condition. Small windows allow detecting sudden changes with a quick operator response but are sensitive to the signal rate change and therefore slow-changing signals are undetected.

Slow and sudden changes in the signal are detected by using two different window sizes, W_1 and W_2. This solutions generates four change signals: $C_t^{W_1,\hat{t}}$, $C_t^{W_2,\hat{t}}$, $C_t^{W_1,t}$, and $C_t^{W_2,t}$ that monitor slow (W_1) or sudden (W_2) increases ($k = \hat{t}$) or decreases ($k = t$) of the uncertainty. Examples of these signals are shown in Fig. 3, bottom.

We represent transitions among tracker conditions based on changes of the uncertainty-based signals to detect global and local changes. The conditions for the global changes, GI_t and GD_t, are defined as:

$$GI_t = \begin{cases} 1 & if \quad C_t^{W_1,\hat{t}} \geq \tau_1 \quad \vee \quad C_t^{W_2,\hat{t}} \geq \tau_1 \\ 0 & otherwise \end{cases} \tag{8}$$

and

$$GD_t = \begin{cases} 1 & if \quad C_t^{W_1,t} \geq \tau_2 \quad \vee \quad C_t^{W_2,t} \geq \tau_2 \\ 0 & otherwise \end{cases} \tag{9}$$

where τ_i $(i = 1, 2)$ represents relative changes (e.g. $\tau_i = 2$ indicates a 200% change).

The proposed tracker-condition FSM model (Figure 2(a)) starts in the *locked-on* state when the tracker is initialized. Then, it passes over to the *scanning* state when a global increase is detected, $GI_t = 1$, and to the *locking-in* state when a (small) sudden uncertainty decrease is detected, $C_t^{W_2,t} > \tau_3$. τ_3 evaluates the amount of decrease change (e.g. $\tau_3 = \tau_2/2$). In the *scanning* state, the FSM passes over to the *locking-in* state when a global decrease is detected, $GD_t = 1$. Then, the FSM maintains its state if there is a sudden uncertainty decrease, $C_t^{W_2,t} > \tau_3$, passes over to the *locked-on* state in case of the stabilization of the uncertainty signal, $C_t^{W_2,t} < \tau_3$, or goes to the *scanning* state if a global uncertainty increase is detected, $GI_t = 1$.

Figure 4 shows an example of temporal segmentation of the tracker condition. The FSM determines the behavior of the tracker when the algorithm follows the target and a wrong object (*locked-on* condition), searching for potential candidates after a tracking failure (*scanning* condition) and focusing on the selected target (*locking-in* condition).

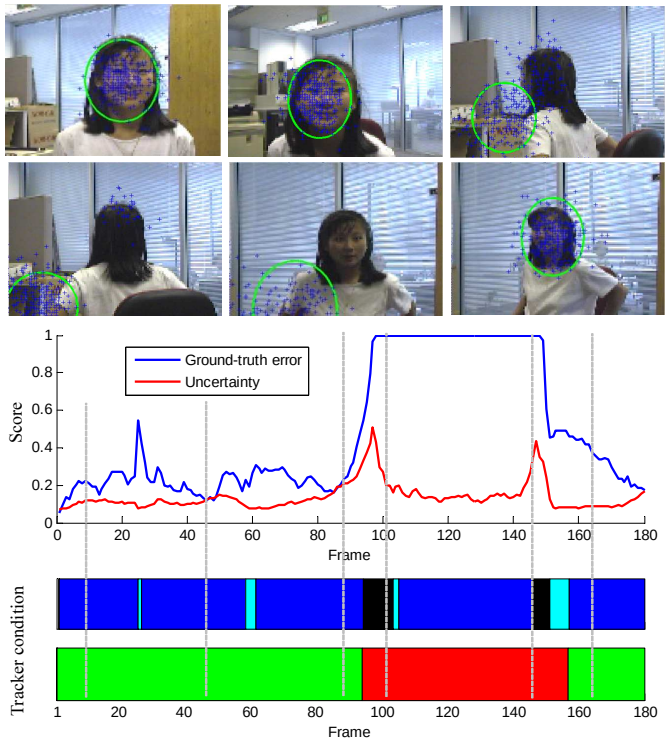


Fig. 4. Sample tracking results, filter uncertainty, ground-truth error, tracker condition estimation and temporal segmentation for the test sequence *seq_mb* (frames 10, 45, 88, 103, 147 and 165). Tracking results and the spatial localization of particles are represented as green ellipses and blue crosses, respectively. The ground-truth error signal was computed as described in Sec. V-B. The color codes for the tracker conditions are as follows. Green: successful tracking; Red: unsuccessful tracking; Black: scanning; Cyan: locking in; Blue: locked on.

D. Detection of recovery from an error

The analysis of the filter uncertainty alone cannot determine a recovery of the tracker after a tracking failure. In fact, locking on a wrong object (distractor) may occur because of similarities between target model and features of other objects in the scene. In this situation, the uncertainty level of the tracker correctly following the target might be the same as the level of the tracker following a distractor (see Fig. 4).

To overcome this limitation and to provide an accurate detection of the recovery after a tracking failure, we propose to use the time-reversibility property [5]. Time reversibility assumes that the movement of an object over time (forward) is also exhibited in the reverse direction and the tracker shall be able to track the target in the reverse (backward) direction. In order to describe the tracking process in forward and reverse direction, let

$$x_t^F = f^F(I_t, x_{t-1}^F, \beta_{t-1}^F) \tag{10}$$

be the state estimated using a forward tracking process and

$$x_t^R = f^R(I_t, x_{t+1}^R, \beta_{t+1}^R) \tag{11}$$

be the state estimated using a reverse tracking process. The superscripts F and R indicate forward and reverse processes and their related variables; x_t, I_t and β_t are target state, current frame and target model at time t, respectively; and f represents the tracking algorithm.

At each time, the recovery analysis is performed when there is a transition from the condition *scanning* to the condition *locked-on*, through the *locking-in* condition. In this case, the time-reversed tracking analysis is started. The reverse tracker is initialized with the current target state estimate (obtained with the forward tracker). A reference point is defined for the reverse analysis (determining its length) as the last known time of the forward tracker estimation when the target state was correctly estimated (successful tracking) before the target was lost (unsuccessful tracking). Therefore, the previously determined values of successful/unsuccessful forward tracking are stored to choose the appropriate reference point. As the forward estimation at the reference point usually contains little information about the target, the real reference point is selected as the furthermost point in the previous half a second that was determined as successful tracking.

Then, the forward and reverse target estimations are compared to detect the recovery after failure. To measure the overlap between two spatial locations (extracted from the target estimations), we use the Dice coefficient [33], which is defined as follows:

$$d_S(x_t^F, x_t^R) = \frac{2\left|A_t^F \cap A_t^R\right|}{\left|A_t^F\right| + \left|A_t^R\right|}, \tag{12}$$

where x_t^F and x_t^R are the forward and reverse target estimations at time t, $\left|A_t^F \cap A_t^R\right|$ is their spatial overlap (in pixels); $\left|A_t^F\right|$ and $\left|A_t^R\right|$ represent their area (in pixels). At each time step t, we detect the error recovery by calculating the distance $d_S(x_{t_0}^F, x_{t_0}^R)$ where t_0 is the reference point to check similarities between forward and reverse tracking estimates; and $x_{t_0}^R$ is obtained by computing the reverse tracking from t to t_0. If the value of $d_S(x_{t_0}^F, x_{t_0}^R)$ is above a certain threshold, τ_4, then the tracker has recovered.

Figure 5 shows two examples of tracking recovery detection. As previously observed in Fig. 4, the uncertainty analysis determined that the tracker became unstable around frames 95-100 and 140-150. Few frames later, the uncertainty stabilized in both cases (Figure 5(a)) recovering from error and (Figure 5(b)) adapting to a wrong object. In both situations, the proposed recovery detection method was able to identify (a) correct and (b) wrong recoveries after the error.

E. Tracker operation condition

Finally, the operation condition during which the tracker is performing successfully or unsuccessfully are defined based on transitions dependent on two conditions, H_1 and H_2 (Fig. 2(b)). Let us assume that the tracker starts from a successful state when it is initialized. Then H_1 is satisfied when the tracker condition moves to or remains in *scanning*. H_2 is satisfied when the tracker condition moves from *locking-in* to *locked-on* and there is a correct recovery from error, i.e. $d_S(x_{t_0}^F, x_{t_0}^R) \geq \tau_4$.

An example of temporal segmentation defining the tracker operation is shown in Fig. 4, bottom. As there are similar objects in the background and the target changes its appearance, the tracker is unable to perform successfully and a failure happens between frames 90 and 160. The temporal

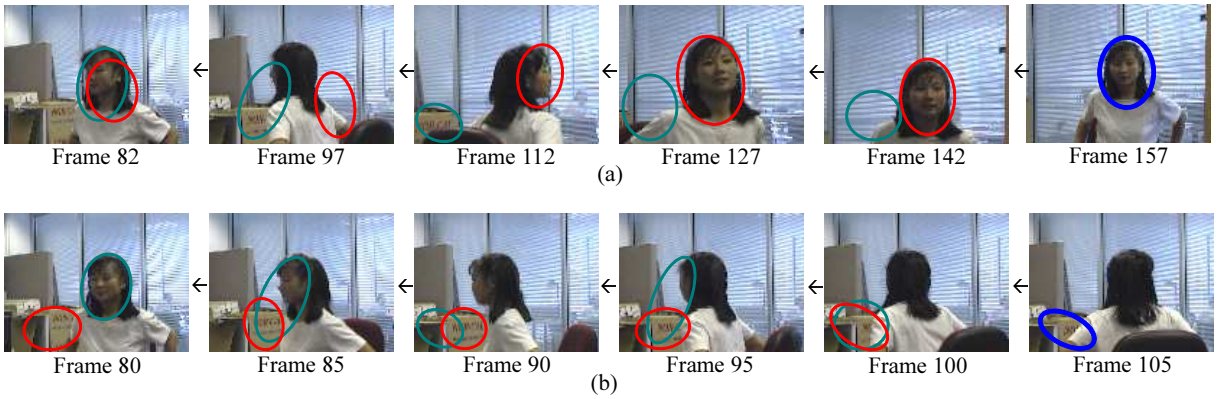


Fig. 5. Examples of reverse tracking applied to detect the recovery from error for the test sequence *seq_mb*. (a) Target recovery after failure in frame 157. (b) Wrong adaptation to a distractor in frame 105. Key. Green ellipse: estimation using forward tracking; Red ellipse: estimation using reverse tracking; Blue ellipse: evaluation of track recovery.

segmentation of successful tracking is correctly performed by combining the tracker condition results and the accurate detection from recovery.

IV. Track-quality estimation

After temporal segmentation of successful and unsuccessful tracking, track quality is estimated for segments during which the tracker is successful. The others segments are considered track-lost segments and therefore discarded for measuring the accuracy of the tracker [1]. We apply the time-reversibility constraint [5] and measure at each time step the similarities between the state estimated with the forward tracker and the state estimated with the reverse tracker (see Eqs. (10) and (11)).

For each evaluation time, a reverse tracker is created and initialized with the current target estimation obtained from the forward tracker (the tracker to be evaluated). Then, tracking is performed in reverse direction until a reference frame defined as the frame where the last successful recovery from error was detected (see Sec. III-D). The initial frame of the video is considered as the first reference frame. Note that the forward and reverse trackers have to be defined using the same tracking algorithm in order to maintain the time-reversibility property.

Then, the differences between the forward and reverse tracker are used to estimate the quality, Q_t, of the current track as:

$$Q_t = 1 - \frac{1}{t - t_1} \sum_{i=t_1}^{t} D(x_i^F, x_i^R), \tag{13}$$

where x_t^F and x_t^R are the target state estimations from the forward and reverse tracking, respectively; t_1 is the reference frame for reverse tracking analysis and $D(.)$ is the function that measures the dissimilarity between forward and reverse analysis. Inspired by the improvement achieved with hybrid evaluation approaches (e.g. [11, 31]), we use a weighted feature combination to generate the dissimilarity measure, $D(.)$:

$$D(x_t^F, x_t^R) = \omega d_S(x_t^F, x_t^R) + (1 - \omega) d_F(x_t^F, x_t^R), \tag{14}$$

where $\omega \in [0, 1]$, $d_S(x_t^F, x_t^R)$ is defined in Eq. (12) and $d_F(x_t^F, x_t^R)$ is a *feature* distance that, for the elliptic color tracker, we define as

$$d_F(x_t^F, x_t^R) = \sqrt{1 - \rho(\mathbf{p}, \mathbf{q})}, \tag{15}$$

where

$$\rho(\mathbf{p}, \mathbf{q}) = \sum_{u=1}^{m} \sqrt{p_u q_u} \tag{16}$$

is the Bhattacharyya coefficient computed between the m–bin color histograms $\mathbf{p}$ and $\mathbf{q}$ of the forward and reverse target estimations. On the one hand, a large value of ω should be selected when there is a high clutter level, because the color histograms of the target will not provide an accurate color representation. Therefore, increasing the weight of the *spatial* distance will increase the performance of the estimated track quality. On the other hand, a small value of ω will increase the weight of the *feature* distance that is useful when the tracker is unable to accurately determine target positions in forward and reverse direction. For generic tracking scenarios, we assume that both distances have an equal impact on track quality and hence $\omega = 0.5$.

An example of track quality estimation is shown in Fig. 6. The progressive decrease in performance (measured with the ground-truth error) is well approximated by the proposed distances. The forward tracking result (depicted as green ellipses) was used to initialize the reverse tracking and to compute the similarity scores. The reverse analysis was performed until the initialization frame of the target (frame 450).

V. Experimental results

A. Experimental setup

We evaluate the results of the proposed approach, ARTE (Adaptive Reverse Tracking Evaluation)[2], and compare it with representative state-of-the-art approaches for empirical standalone quality evaluation: Observation Likelihood (OL) [6], covariance of the target state (SU) [7], frame-by-frame reverse-tracking evaluation using template inverse matching

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

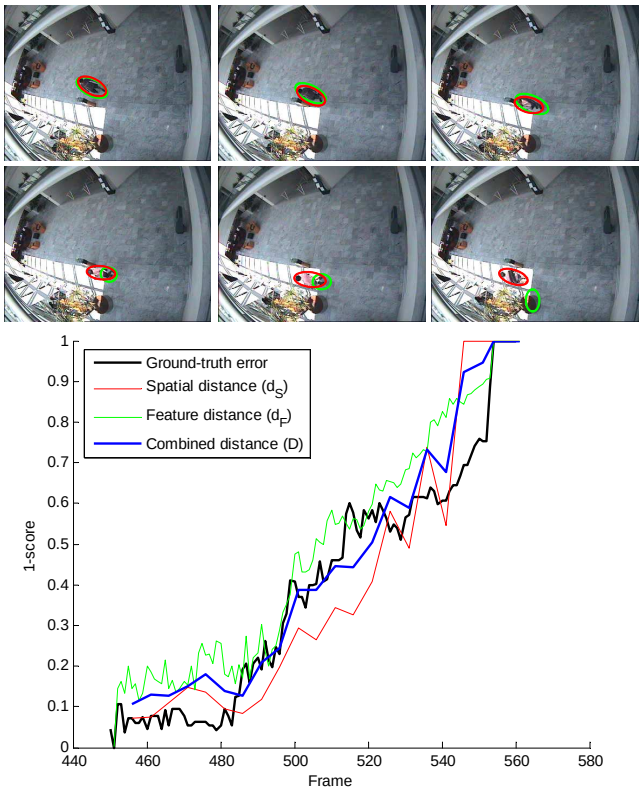


Fig. 6. Comparison of proposed distances to measure track quality. The sample images correspond to frames 460, 480, 500, 520, 540 and 560. Tracking results and ground-truth data are represented as green and red ellipses, respectively. The ground-truth error is measured as the spatial overlap between estimated and ground-truth target (Eq. 12).

(TIM) [18] and full-length reverse-tracking evaluation using the same tracking algorithm (FBF) [5].

The evaluation dataset is composed of sequences from CAVIAR[3], PETS2001[4], PETS2010[5], CLEMSON[6] and VISOR[7] datasets. These sequences present challenging situations for tracking such as total or partial occlusions, clutter, and illumination or scale changes (Table II). The initialization of each target is shown in Fig. 7. To evaluate the performance, we use ground-truth data consisting of the ellipse best fitting the target at each frame and described with its centroid, axes and rotation angle.

The parameters of the tracker [32] are the same for all the targets. Color histograms are generated in the RGB space for pedestrian targets (P) and in the HSV space for face targets (F), using 8x8x8 bins in both cases; $\sigma_{x,y} = 5$, $\sigma_{H_x,H_y} = 0.75$, $\sigma_\theta = 4°$, $\sigma_c = 0.2$; 300 samples/particles are used in the experiments. The values of $\tau_1 = 2$, $\tau_2 = -2.5$, $\tau_3 = -1.25$ and $\tau_4 = 0.5$. Due to the statistically nature of the particle filter, we run the tracker 10 times for each sequence.

[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/

Table II
DESCRIPTION OF THE EVALUATION DATASET (KEY. SC: SCALE CHANGES. AC: APPEARANCE CHANGES. IC: ILLUMINATION CHANGES. O:OCCLUSIONS. C:CLUTTER.)

Dataset	Target	Size	Characteristics
CAVIAR	P1 – P4	384x288	IC, C
PETS2001	P5 – P10	768x576	SC, O, C
PETS2010	P12 – P18	768x576	O, C
D1	F1 – F4	128x196	SC, AC, C, O
VISOR	F5, F6	352x288	SC, C, O

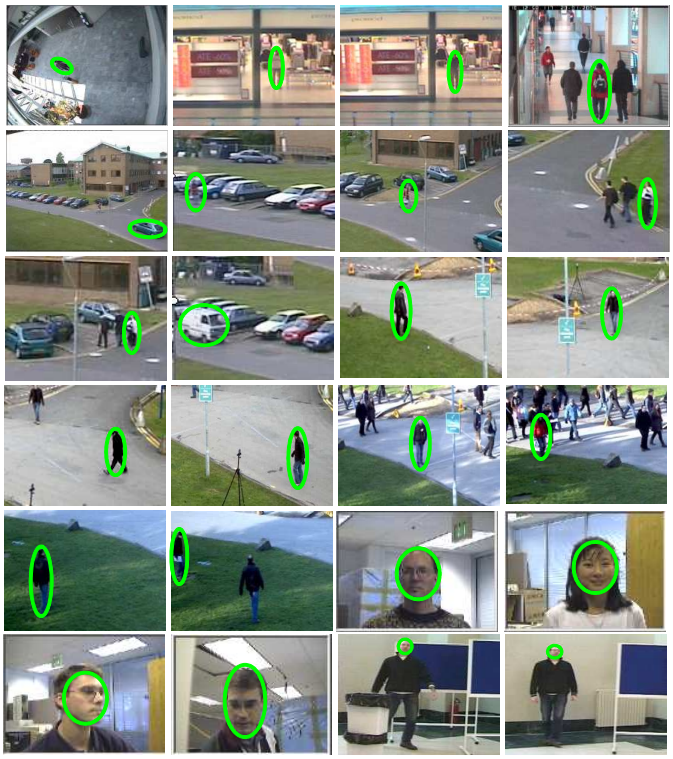


Fig. 7. Target initialization for the evaluation dataset. From top-left to bottom-right: Pedestrian targets: *Browse_WhileWaiting1* (P1), *OneLeaveShopReenter1front* (P2), *OneLeaveShopReenter2front* (P3), *ThreePastShop2cor* (P4), *Camera1_testing* (P5–P10), *S2_L1_view001* (P11–P14), *S2_L2_view0001* (P15, P16) and *S2_L3_view001* (P17, P18); face targets: *seq_bb* (F1), *seq_mb* (F2), *seq_sb* (F3), *seq_villains2* (F4), *occlusion_1* (F5) and *occlusion_2* (F6).

B. Performance evaluation criteria

The error between the tracking data and the ground-truth data is quantified using the spatial overlap of the corresponding target areas (Eq. (12)). Low performance is indicated by values close to 0 (i.e. small overlap). High performance is indicated by values close to 1 (i.e. large overlap). Track quality is evaluated once every five frames.

The performance of temporal segmentation (see Sec. III) is evaluated using Receiver Operating Characteristic (ROC) analysis. This analysis requires the definition of ground-truth segmentation. A successful (unsuccessful) track is determined when the error measure $d_S(x_t^e, x_t^g)$, defined as in Eq. (12), is larger (smaller) than the minimum allowed overlap between x_t^g, the ground truth, and x_t^e, the estimation.

Finally, the performance of the proposed approach is evaluated by a correlation analysis against the error measure for the case of successful tracking (determined using the proposed

Table III
COMPARATIVE RESULTS. ROC ANALYSIS USING 10 RUNS EXPRESSED AS AVERAGE ± STANDARD DEVIATION. (KEY. AUC: AREA UNDER THE CURVE, FPR: FALSE POSITIVE RATE, TPR: TRUE POSITIVE RATE)

Approach	AUC	FPR	TPR
ARTE	$.87 \pm .02$	$.16 \pm .01$	$.89 \pm .03$
OL [6]	$.66 \pm .07$	$.37 \pm .05$	$.61 \pm .11$
SU [7]	$.76 \pm .04$	$.38 \pm .03$	$.81 \pm .06$
TIM [18]	$.44 \pm .01$	$.28 \pm .02$	$.27 \pm .04$
FBF [5]	$.87 \pm .03$	$.25 \pm .03$	$.95 \pm .02$

approach). We use the Pearson product-moment correlation coefficient [34] between ground-truth and estimated data:

$$\rho = \left| \frac{E\left[(X^e - \mu_{X^e})(X^g - \mu_{X^g})\right]}{\sigma_e \sigma_g} \right|, \qquad (17)$$

where X^e and X^g represent the estimated and ground-truth data for each video sequence; μ_{X^e} and μ_{X^g} represent their respective means; σ_{X^e} and σ_{X^g} represent their respective standard deviations; $\rho \in [0, 1]$, with values close to 1 indicating high correlation with ground-truth data.

C. Temporal segmentation to detect successful tracking

The results regarding the temporal segmentation between successful and unsuccessful tracking are summarized in Fig. 8 and Table III. Feature-based measures (for OL and SU) demonstrated their dependence on the clutter level (for OL) and on the adaptation to wrong targets (OL and SU), thus obtaining intermediate results. SU obtained better results as it relies on filter uncertainty. On the one hand, TIM demonstrated the inaccuracy of short-length reverse-based evaluation due to the adaptation of the measure to tracking errors. On the other hand, the time-reversibility property is useful to segment correct tracking and, in fact, the full-length reverse-based evaluation (FBF) obtained high performance. ARTE obtained similar AUC compared to FBF. An intersection of both ROC curves shows that FBF outperforms ARTE with higher *true positive rate*. However, the observed *false alarm rate* for FBF is also larger than that for ARTE and ARTE obtained better *true positive rate* than FBF in the case of low *false alarm rate*. In addition to this, the execution time of ARTE is considerable lower than that of FBF: approximately 50 (10) times without (with) track quality estimation. The temporal segmentation allowed determining adaptively the reference points for reverse analysis whilst in FBF this point is fixed (the initialization frame of the target). FBF has an exponential increase of computational cost and therefore it is inadequate to evaluate trackers on long sequences.

Figure 9 shows two temporal segmentation examples. Figure 9(a) illustrates a case of failure and wrong target adaptation: the tracker starts in the *locked-on* condition and then moves to an over-illuminated area. Here the tracker loses the target (tracker condition: *scanning*). Few frames later, the tracker is distracted by a background object (tracker condition *locking-in*). Finally, the tracker is completely adapted to the wrong object (tracker condition: *locked-on*). A reverse tracking analysis is performed to check the correct recovery after error and fails indicating the wrong target adaptation.

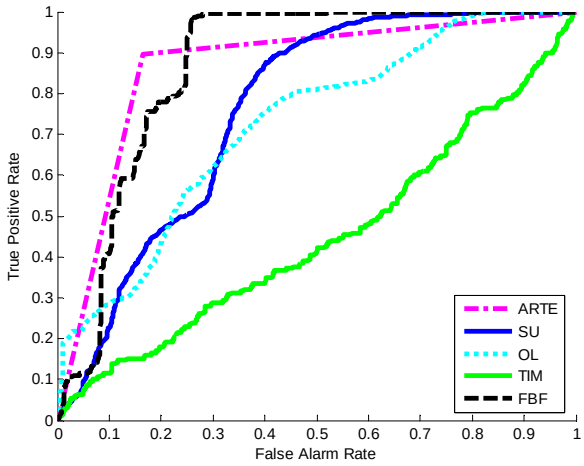


Fig. 8. ROC curves for the segmentation between successful and unsuccessful tracking using the evaluation dataset. (Key. ARTE: proposed approach, OL: Observation Likelihood [6], SU: Covariance of the target state [7], TIM: frame-by-frame reverse tracking [18]; FBF: full reverse tracking [5]).

A second temporal segmentation example with failure recovery is shown in Fig. 9(b). A moving head is tracked (tracker condition: *locked-on*) until it gets occluded by a blackboard (tracker condition: *scanning*). Then, the tracker recovers the correct target (tracker conditions: *locking-in* and *locked-on*). Successful tracker recovery is verified by the reverse-based proposed method. Then, the target is again lost due to a quick movement (tracker condition: *scanning*) and recovered a few frames later (tracker conditions: *locking-in* and *locked-on*). Finally, a target estimation refinement happens as the filter uncertainty is decreased (transitions between *locking-in* and *locked-on* conditions).

Note that, although the proposed approach shares with the Expected-Log Likelihood (ELL) statistic [35] the generic idea of measuring dissimilarities between prior and posterior distributions for particle-filter-based abnormality detection, it differs by considering this prior conditioned to the observed data and by detecting slow and sudden changes (ELL is only valid for slow ones [6]). Moreover, the filter uncertainty is not sufficient to estimate the tracker operation condition as data can be consistent due to distractors. Hence, the use of a statistic such as ELL will not be able to provide this condition.

D. Track-quality estimation

Results comparing the correlation between track quality estimators and ground-truth data are summarized for pedestrian and face targets in Table IV. As for pedestrians, ARTE achieved an average ground-truth correlation of 57.3% whilst the other approaches obtained 50.5% (OL), 48.0% (SU), 11.3% (TIM) and 33.15% (FBF) correlation. OL obtained varying correlation values showing its dependency to wrong target adaptation and to different levels of clutter. The first situation can be easily observed for P2, P13 and P17, whilst the second situation is observed for P6 and P7 (where there is no tracking failure). In particular, high performance was achieved in case of correct tracking or failures due to target–

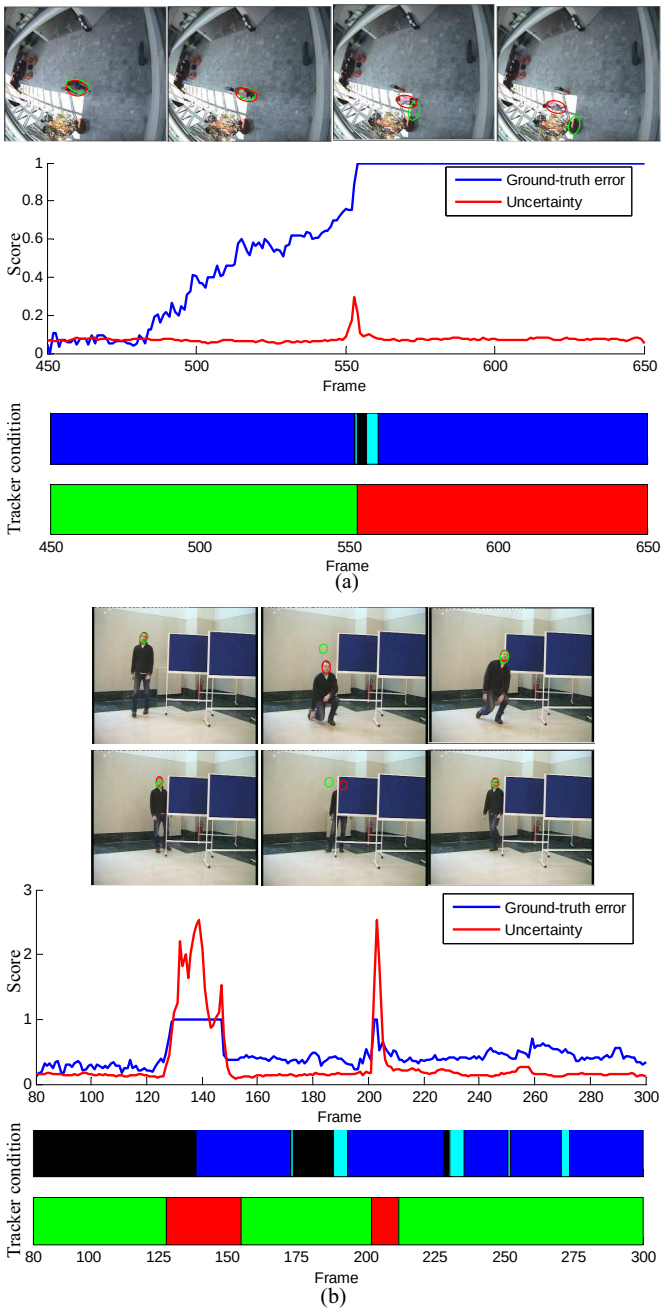


Fig. 9. Sample tracking results, tracker condition estimation and temporal segmentation for (a) target P1 (*Browse_WhileWaiting1* sequence; frames shown are 525, 540, 560 and 580) and (b) target H5 (*occlusion_1* sequence; frames shown are 115, 140, 160, 180, 210 and 225). Tracking results and ground-truth data are represented as green and red ellipses, respectively. The color codes for the tracker conditions are as follows. Green: successful tracking; Red: unsuccessful tracking; Black: scanning; Cyan: locking in; Blue: locked on.

model dissimilarities. OL obtained the best results, thus confirming the conclusions of [12]. SU obtained low performance showing a high dependency on wrong target adaptation (P1, P2, P3, P7 and P17) and being unable to evaluate track quality (P10) as the particle filter tried to keep it constant during the analysis. However, high performance was obtained for cases without track-quality degradation (P5) or failure without wrong adaptation or recovery (P8). TIM achieved

Table IV
COMPARISON OF TRACK QUALITY ESTIMATION PERFORMANCE FOR PEDESTRIAN (P1-P18) AND FACE TARGETS (F1-F6). BOLD INDICATES THE BEST RESULT FOR EACH TARGET. (KEY. ARTE: PROPOSED APPROACH, OL: OBSERVATION LIKELIHOOD [6], SU: COVARIANCE OF THE TARGET STATE [7], TIM: FRAME-BY-FRAME REVERSE TRACKING [18]; FBF: FULL REVERSE TRACKING [5])

Target	Pearson correlation coefficient				
	OL	SU	TIM	FBF	ARTE
P1	$\mathbf{.86 \pm .04}$	$.25 \pm .06$	$.07 \pm .05$	$.09 \pm .05$	$.75 \pm .15$
P2	$.34 \pm .10$	$.20 \pm .09$	$.10 \pm .16$	$.15 \pm .07$	$\mathbf{.45 \pm .20}$
P3	$.20 \pm .06$	$.49 \pm .08$	$.10 \pm .04$	$.02 \pm .12$	$\mathbf{.83 \pm .11}$
P4	$\mathbf{.64 \pm .05}$	$.53 \pm .06$	$.10 \pm .01$	$.33 \pm .17$	$.46 \pm .06$
P5	$.95 \pm .02$	$\mathbf{.96 \pm .02}$	$.08 \pm .04$	$.95 \pm .03$	$.86 \pm .10$
P6	$.44 \pm .10$	$\mathbf{.56 \pm .08}$	$.05 \pm .03$	$.45 \pm .20$	$.47 \pm .19$
P7	$.24 \pm .15$	$.25 \pm .11$	$.05 \pm .02$	$.17 \pm .09$	$\mathbf{.44 \pm .14}$
P8	$.54 \pm .13$	$\mathbf{.71 \pm .16}$	$.08 \pm .07$	$.47 \pm .21$	$.48 \pm .05$
P9	$\mathbf{.57 \pm .14}$	$.54 \pm .14$	$.15 \pm .11$	$.10 \pm .09$	$.55 \pm .10$
P10	$.11 \pm .03$	$.23 \pm .08$	$.12 \pm .12$	$.02 \pm .01$	$\mathbf{.40 \pm .10}$
P11	$.81 \pm .15$	$.64 \pm .11$	$.32 \pm .09$	$.58 \pm .11$	$\mathbf{.98 \pm .01}$
P12	$.59 \pm .11$	$.28 \pm .05$	$.06 \pm .05$	$.27 \pm .09$	$\mathbf{.75 \pm .08}$
P13	$.28 \pm .10$	$\mathbf{.52 \pm .14}$	$.10 \pm .13$	$.14 \pm .05$	$.44 \pm .15$
P14	$.25 \pm .05$	$.66 \pm .12$	$.17 \pm .03$	$.71 \pm .13$	$\mathbf{.77 \pm .04}$
P15	$\mathbf{.59 \pm .08}$	$.32 \pm .05$	$.25 \pm .12$	$.43 \pm .07$	$.48 \pm .18$
P16	$.64 \pm .11$	$.45 \pm .10$	$.11 \pm .06$	$.16 \pm .03$	$\mathbf{.75 \pm .05}$
P17	$.20 \pm .05$	$.18 \pm .04$	$.20 \pm .10$	$.34 \pm .09$	$\mathbf{.42 \pm .08}$
P18	$.80 \pm .14$	$.82 \pm .09$	$.24 \pm .12$	$.26 \pm .12$	$\mathbf{.87 \pm .15}$
mean	$.50 \pm .10$	$.48 \pm .08$	$.11 \pm .09$	$.33 \pm .10$	$\mathbf{.57 \pm .03}$
F1	$.19 \pm .06$	$.60 \pm .11$	$.12 \pm .04$	$.44 \pm .11$	$\mathbf{.74 \pm .17}$
F2	$.63 \pm .15$	$.38 \pm .26$	$.07 \pm .05$	$.24 \pm .09$	$\mathbf{.65 \pm .10}$
F3	$.20 \pm .17$	$.57 \pm .15$	$.06 \pm .06$	$\mathbf{.79 \pm .13}$	$.34 \pm .06$
F4	$.14 \pm .09$	$\mathbf{.65 \pm .04}$	$.07 \pm .03$	$.25 \pm .08$	$.37 \pm .09$
F5	$\mathbf{.71 \pm .05}$	$.63 \pm .04$	$.08 \pm .04$	$.42 \pm .15$	$\mathbf{.71 \pm .08}$
F6	$.31 \pm .11$	$.17 \pm .10$	$.09 \pm .05$	$.05 \pm .03$	$\mathbf{.32 \pm .21}$
mean	$.36 \pm .12$	$.50 \pm .09$	$.08 \pm .05$	$.37 \pm .11$	$\mathbf{.52 \pm .12}$
tot mean	$.47 \pm .08$	$.48 \pm .08$	$.09 \pm .08$	$.34 \pm .10$	$\mathbf{.56 \pm .07}$

the worst results as the use of short-length reverse analysis accumulates errors over time. A few frames after a tracking failure, TIM was unable to evaluate tracking failure as this approach adapts to the target estimation in short temporal windows. FBF also presented varying results and obtained high performance for P5 and P14; intermediate performance for P4, P6, P8, P11 and P15; and low performance for P1, P3, P9, P10, P13 and P16. The reason for this behavior is due to the metric applied, the Mahalanobis distance, which does not work well with different degradations of track quality. This distance measures data similarity considering their means and covariance matrix. However, probabilistic tracking usually provides weighted estimations of target state (e.g. particle filter weights). Therefore, small changes in the covariance matrix of the target state are not measured by the Mahalanobis distance. Moreover, this distance has not got a fixed range of values that identifies tracking failure without ambiguities. Hence, several Mahalanobis distance values can correspond to a tracking failure and their correlation with ground-truth data is low. In addition to this, the track quality is computed using only the last estimated state of the reverse analysis (performed until the reference frame). Hence, there is an information loss due to the not-computed reverse-forward comparisons. ARTE addresses all these issues achieving a good trade-off in all the test sequences.

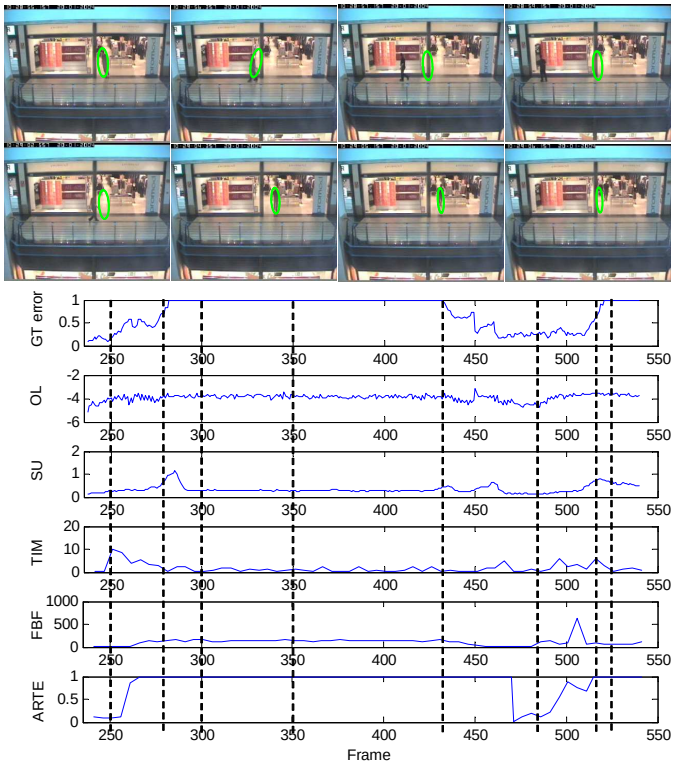
As for face targets, ARTE achieved an average ground-truth correlation of 52.8% whilst other approaches obtained 36.4% (OL), 50.3% (SU), 8.7% (TIM) and 37.7% (FBF) correlation. A decrease in performance compared to the pedestrian results can be observed that is due to the higher complexity of face targets. OL and SU obtained intermediate results (similar to ARTE) compared to the pedestrian target results. This performance can be explained with the initialization process and the type of sequences. A face target is easier to annotate than a pedestrian target and the HSV color space offers a good description of face targets. Hence, the corresponding target model is more accurate for faces than for pedestrians. Moreover, a common tracking error is due to occlusion with an object whose appearance is very different from that of the target and therefore tracker recovery was successful in most cases. In this situation, OL and SU increased their performance as they depend on similarities between target model and candidates. TIM and FBF obtained low performance due to error accumulation for short-length approaches (TIM) or an inappropriate metric used (FBF), as commented earlier.

Sample results of track quality estimation are shown for wrong target adaptation and correct target recovery after a tracking failure in Fig. 10. Figure 10(a) shows how the tracker loses the target and adapts to the most similar background patches. Then, it first recovers the target and then loses it at the end of the sequence. Figure 10(b) shows the tracking of a moving head that gets occluded twice by another moving head and by a blackboard. The first occlusion was by a similar target model (not detected by OL) and the second occlusion was due to a model dissimilarity (correctly detected by all approaches). In this case, track quality is correctly estimated by ARTE only.

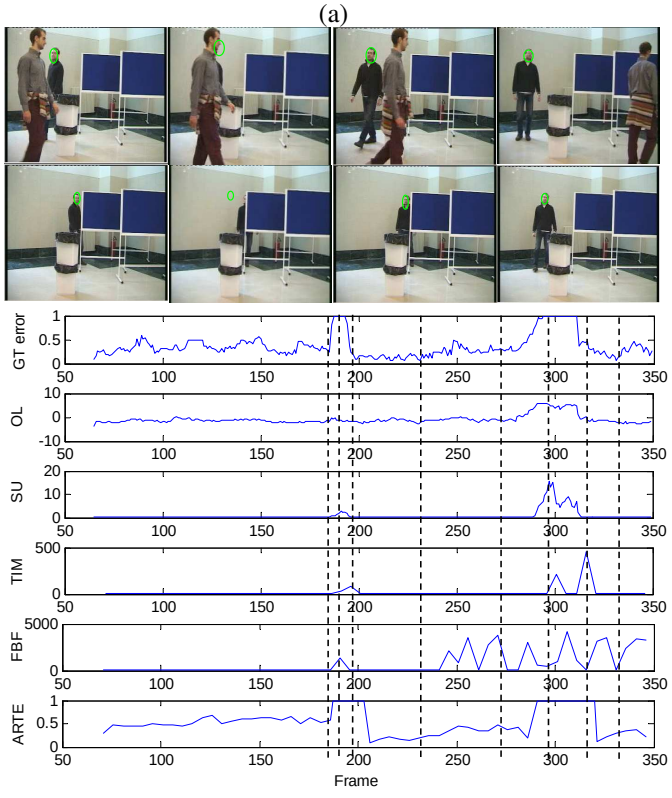
As final remark, ARTE detects a recovery few frames later than it actually happened, as shown by the above examples. This latency is due to the filter uncertainty operator that decreases because it integrates previous values. This latency is due to the delayed detection and can be overcome by changing the size of the temporal window: a performance increase (in terms of ground-truth correlation) is expected with a post-processing stage. Nevertheless, a delay in the output of ARTE is introduced to allow the re-calculation of track quality estimations when a recovery is detected. We have decided not to perform this post-processing to avoid an unfair comparison with the state-of-the-art methods as they produce the track-quality data without any post-processing.

VI. Conclusions and Future Work

We have presented a track quality estimator in the absence of ground-truth data. A novel adaptive analysis strategy based on the uncertainty of the tracking filter and the time-reversibility constraint has been proposed. Tracking failures are identified by analyzing changes of the filter uncertainty. Time-reversibility is applied to check recovery after a tracking failure. Then, track quality is estimated for successful tracking cases by using a reverse tracking analysis that is based on similarities between reverse and forward tracking in terms of color and spatial distances. Experimental results over a



(a)



(b)

Fig. 10. Sample tracking results, ground-truth error and track quality estimators for (a) target P3 (frames shown are 250, 280, 300, 350, 435, 480, 510 and 525) and (b) target H6 (frames shown are 180, 185, 195, 230, 275, 295, 310 and 335). The methods under analysis are the proposed approach (ARTE), Observation Likelihood (OL) [6], Covariance of the target state (SU) [7], frame-by-frame reverse tracking (TIM) [18] and full reverse tracking (FBF) [5]. Tracking results are shown as green ellipses.

heterogeneous dataset showed that the proposed approach outperforms state-of-the-art algorithms. The approach was demonstrated in a particle filter framework and is applicable to multi-hypothesis trackers that use some forms of uncertainty related to the spread of the generated hypotheses. Its application to single-hypothesis trackers not based on Bayesian filtering requires an adaptation of the algorithm output, for example with a transformation that computes a correlation map for target location [27]. Other future research directions include investigating adaptive thresholding techniques and the fusion of multiple trackers based on track quality.

REFERENCES

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

[2] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data,metrics, and protocol," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31(2), 2008, pp. 319 –336.

[3] J. Black, T. Ellis, and P. Rosin, "A novel method for video tracking performance evaluation," in *Proc. of IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, Nice (France), October 2003, pp. 125–132.

[4] D. Chau, F. Bremond, and M. Thonnat, "Online evaluation of tracking algorithm performance," in *Proc. of Int. Conf. on Imaging for Crime Prevention and Detection*. London (UK), December 2009.

[5] H. Wu, A. Sankaranarayanan, and R. Chellappa, "Online empirical evaluation of tracking algorithms," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32(8), 2010, pp. 1443–1458.

[6] N. Vaswani, "Additive change detection in nonlinear systems with unknown change parameters," in *IEEE Trans. on Signal Processing*, vol. 55(3), 2007, pp. 859–872.

[7] E. Maggio, F. Smerladi, and A. Cavallaro, "Adaptive multifeature tracking in a particle filtering framework,," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17(10), October 2007, pp. 1348–1359.

[8] F. Van der Heijen, "Consistency checks for particle filters," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 1, no. 1, 2006, pp. 140–145.

[9] E. Erdem, Sankur, and A. Tekalp, "Performance measures for video object segmentation and tracking," in *IEEE Trans. on Image Processing*, vol. 13(7), 2004, pp. 937–951.

[10] R. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27(10), 2005, pp. 1631–1643.

[11] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *Proc. of IEEE Int. Conf. on Pattern Recognition*, Istambul (Turkey), 23-26 Aug. 2010, pp. 2756–2759.

[12] J. SanMiguel, A. Cavallaro, and J. Martinez, "Evaluation of online quality estimators for video object tracking," in *IEEE Int. Conf. on Image Processing*, Hong Kong (China), 26-29 Sept. 2010, pp. 4657–4660.

[13] C. Piciarelli, G. Foresti, and L. Snidaro, "Trajectory clustering and its applications for video surveillance," in *Proc. of IEEE Advanced Video-based Signal Surveillance*, Como (Italy), 15-16 September 2005, pp. 40–45.

[14] D. Hall, "Automatic parameter regulation of perceptual systems," in *Image and Vision Computing*, vol. 24(8), 2006, pp. 870–881.

[15] H. Wu and Q. Zheng, "Self-evaluation of visual tracking systems," in *Proc. of Army Science Conf.*, Orlando (FL, USA), 29 Nov.-2 Dec. 2004.

[16] A. Doulamis, "Dynamic tracking re-adjustment: a method for automatic tracking recovery in complex visual environments," in *Multimedia Tools and Applications*, vol. 50(1), 2010, pp. 49–73.

[17] E. Polat, M. Yeasin, and R. Sharma, "Tracking body parts of multiple people: a new approach," in *Proc. of the IEEE Workshop on Multi-Object Tracking*, Vancouver (Canada), August 2001, pp. 35–42.

[18] R. Liu, S. Li, X. Yuan, and R. He, "Online determination of track loss using template inverse matching," in *Proc. of the Int. Workshop on Visual Surveillance*, Marseille (France), 17 October 2008.

[19] P. Pan, F. Porikli, and D. Schonfeld, "Recurrent tracking using multifold consistency," in *Proc. of IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, Miami (USA), 20-25 June 2009.

[20] Z. Han, Q. Ye, and J. Jiao, "Online feature evaluation for object tracking using kalman filter," in *Proc. of the IEEE Int. Conf. on Pattern Recognition*, Tampa (FL, USA), 8-11 Dec. 2008, pp. 1–4.

[21] E. Erdem, A. Tekalp, and B. Sankur, "Video object tracking with feedback of performance measures,," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4(4), 2004, pp. 310–324.

[22] P. Correia and F. Pereira, "Stand-alone object segmentation quality evaluation," in *EURASIP Journal on Applied Signal Processing*, vol. 4, 2002, pp. 389–400.

[23] C. Motamed, "Motion detection and tracking using belief indicators for an automatic visual-surveillance system," in *Image and Vision Computing*, vol. 24(11), 2006, pp. 1192–1201.

[24] K. Nickels and S. Hutchinson, "Estimating uncertainty in ssd-based feature tracking," in *Image and Vision Computing*, vol. 20(1), 2002, pp. 47–58.

[25] E. Loutas, I. Pitas, and C. Nikou, "Entropy-based metrics for the analysis of partial and total occlusion in video object tracking," in *IEE Proc. - Vision, Image, and Signal Processing*, vol. 151(6), 2004, pp. 487–497.

[26] L. Lu, X. Dai, and G. Hager, "A particle filter without dynamics for robust 3d face tracking," in *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition Workshop*, vol. 5, Washington (DC, USA), 27 June- 2 July 2004, pp. 70–73.

[27] V. Badrinarayanan, P. Perez, F. Le Clerc, and L. Oisel, "On uncertainties, random features and object tracking," in *Proc. of IEEE Int. Conf. on Image Processing*, San Antonio (TX, USA), 16-19 Sept. 2007, pp. 61–64.

[28] A. Bagdanov, A. Del-Bimbo, F. Dini, and W. Nunziati, "Adaptive uncertainty estimation for particle filter-based trackers," in *Proc. of Int. Conf. on Image Analysis and Processing*, Modena (Italy), 10-14 Sept. 2007, pp. 331–336.

[29] V. Badrinarayanan, P. Perez, F. Le Clerc, and L. Oisel, "Probabilistic color and adaptive multi-feature tracking with dynamically switched priority between cues," in *Proc. of IEEE Int. Conf. on Computer Vision*, Rio de Janeiro (Brasil), 14-21 October 2007, pp. 1–8.

[30] R. M. Powers and L. Y. Pao, "Power and robustness of a track-loss detector based on kolmogorov-smirnov tests," in *Proc. of American Control Conf.*, Minneapolis (MN, USA), 14-16 June 2006, pp. 3757–3764.

[31] P. Pan, F. Porikli, and D. Schonfeld, "A new method for tracking performance evaluation based on reflective model and perturbation analysis," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Taipei (Taiwan), 19-24 April 2009, pp. 3529–3532.

[32] K. Nummiaro, E. Koller-Meier, and E. Van Gool, "An adaptive colour-based particle filter," in *Image and Vision Computing*, vol. 21(1), 2003, pp. 99–110.

[33] A. Nghiem, F. Bremond, M. Thonnat, and V. Valentin, "Etiseo,

performance evaluation for video surveillance systems," in *IEEE Conf. on Advanced Video and Signal Based Surveillance*, London (UK), 5-7 Sept. 2007, pp. 476–481.

[34] P. Chen and P. Popovich, *Correlation: Parametric and non-parametric measures*. Thousand Oaks, 2002.

[35] N. Vaswani and R. Chellappa, "A particle filtering approach to abnormality detection in nonlinear systems and its application to abnormal activity detection," in *Proc. of Int. Workshop on Statistical and Computational Theories of Vision*, Nice (France), 13-16 Oct. 2003, pp. 1–8.

Juan C. SanMiguel received the M.S. degree in Electrical Engineering ('Ingeniero de Telecomunicación' degree) in 2006 and the Ph.D. in Computer Science and Telecommunication in 2011, both at Universidad Autónoma de Madrid (Spain). Since 2005, he has been working as a researcher in the Video Processing and Understanding Lab (VPU-Lab) at Universidad Autónoma of Madrid, where he has been teaching assistant in the Telecommunications degree for the courses "Linear Systems" and "Advanced topics in signal processing". During this period, he has participated in several national projects and colaborations with the private sector related with multimedia content analysis, description and transmission. He also serves as a reviewer for several International Journals and Conferences. His current research interests are focused on the video analysis domain including object tracking, activity recognition, performance evaluation and real-time systems. He has co-authored 15 publications in international conferences and journals.

Andrea Cavallaro received the Laurea (summa cum laude) degree from the University of Trieste, Italy, in 1996, and the Ph.D. degree from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 2002, both in electrical engineering. In 1996 and 1998, he served as a Research Consultant with the Image Processing Laboratory, University of Trieste, Italy, working on compression algorithms for very low bitrate video coding. From 1998 to 2003, he was a Research Assistant with the Signal Processing Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. Since 2003, he has been with Queen Mary University of London, UK, where he is Professor of Multimedia Signal Processing. Prof. Cavallaro has authored more than 130 papers, including ten book chapters and two books, Multi-Camera Networks (Elsevier, 2009) and Video Tracking (Wiley, 2011). He was awarded a Research Fellowship with British Telecommunications (BT) in 2004/2005, three Student Paper Awards at IEEE ICASSP in 2005, 2007, and 2009, and the Best Paper Award at IEEE AVSS 2009. He is an elected member of the IEEE Image, Video, and Multidimensional Signal Processing Technical Committee; he served as Technical Chair for IEEE AVSS 2011; WIAMIS 2010 and EUSIPCO 2008; and as General Chair for IEEE/ACM ICDSC 2009, BMVC 2009, and IEEE AVSS 2007. Prof. Cavallaro is Area Editor for IEEE Signal Processing Magazine and Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON SIGNAL PROCESSING.

José M. Martínez received the Ingeniero de Telecomunicación degree (six years engineering program) in 1991 and the Doctor Ingeniero de Telecomunicación degree (PhD in Communications) in 1998, both from the E.T.S. Ingenieros de Telecomunicación of the Universidad Politécnica de Madrid. He is Associate Professor at the Escuela Politécnica Superior of the Universidad Autónoma de Madrid. His professional interests cover different aspects of advanced video surveillance systems and multimedia information systems. Besides his participation in several Spanish national projects (both with public and private funding), he has been actively involved in European projects dealing with multimedia information systems applied to the cultural heritage (e.g., RACE 1078 EMN, European Museum Network; RACE 2043 RAMA, Remote Access to Museums Archives; ICT-PSP-FP7-250527 ASSETS, Advanced Search Services and Enhanced Technological Solutions for the Europeana Digital Library), education (e.g., ET 1024 TRENDS, Training Educators Through Networks and Distributed Systems), multimedia archives (e.g., ACTS 361 HYPERMEDIA, Continuous Audiovisual Digital Market in Europe) and semantic multimedia networked systems (e.g., IST FP6-001765 acemedia, IST FP6-027685 Mesh). He is author and co-author of more than 100 papers in international journals and conferences, and co-author of the first book about the MPEG-7 Standard published 2002. He has acted as auditor and reviewer for the EC for projects of the frameworks program for research in Information Society and Technology (IST). He has acted as reviewer for journals and conferences, and has been Technical Co-chair of the International Workshop VLBV'03, Special Sessions Chair of the International Conference SAMT 2006, Special Sessions Chair of the 9th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 2008, Program co-chair of the 7th International Workshop on Content-based Multimedia Indexing CBMI 2009 and General chair of the 9th International Workshop on Content-based Multimedia Indexing CBMI 2011.