Adaptive multi-feature tracking in a particle filtering framework

Emilio Maggio, *Student Member, IEEE*, Fabrizio Smeraldi, Andrea Cavallaro, *Member, IEEE*

***Abstract*— We propose a tracking algorithm based on an adaptive multi-feature statistical target model. The features are combined in a single particle filter by weighting their contributions using a novel reliability measure derived from the particle distribution in the state space. This measure estimates the reliability of the information by measuring the spatial uncertainty of features. A modified resampling strategy is also devised to account for the needs of the feature reliability estimation. We demonstrate the algorithm using color and orientation features. Color is described with part-wise normalized histograms. Orientation is described with histograms of the gradient directions that represent the shape and the internal edges of a target. A feedback from the state estimation is used to align the orientation histograms as well as to adapt the scales of the filters to compute the gradient. Experimental results over a set of real-world sequences show that the proposed feature weighting procedure outperforms state-of-the-art solutions and that the proposed adaptive multi-feature tracker improves the reliability of the target estimate while eliminating the need of manually selecting each feature's relevance.**

***Index Terms*— Particle filter, multi-feature, representation, tracking, color histogram, orientation histogram, feature reliability.**

I. Introduction

VIDEO-BASED trackers are important components in many applications, such as video surveillance, medical image sequence analysis, augmented reality, smart rooms, and object-based video compression. Tracking algorithms aim to estimate the position (and the shape) of a target over time. To this end, a target model is first defined and then searched for in subsequent frames using a function that evaluates the similarity between the model and a candidate. A critical issue is the distinctiveness of the target model with respect to the background and clutter.

A. Multi-feature target representations

A common solution to improve the target model distinctiveness is to use multiple features, such as color and edges. Color is widely used for target representation to perform the data association task [1], [2]. Color histograms have been used in the mean-shift algorithm for gradient descent search [3] and in particle filtering for likelihood estimation [4]. Color histograms allow significant data reduction, are robust to partial occlusions and can be computed efficiently. However, their descriptiveness is limited by the lack of spatial information, which makes it difficult to discriminate between targets with similar color properties. To reduce this problem, spatial information can be introduced by using multiple localized histograms over semi-overlapping areas [5] or by associating with each color bin the first two spatial moments of the pixel coordinates of the corresponding color [6]. However, despite the inclusion of spatial information, a target model defined by color histograms only can still be misled by changes in scene illumination, by out-of-plane object rotations, and by background clutter. For this reason, gradient information can be used to complement color information [7], [8]. As the gradient is usually computed on the luminance information, the edges of an object do not depend on the chromatic content; thus the tracker can exploit the complementary information. Existing representations usually discard the edge information inside a target [7], [8]. The projection of the gradient on the target border is used in face tracking [7]. Similarly, edge density near the target border can be computed using a binary Laplacian map [8]. More detailed edge information obtained from the histogram of the gradient orientation is used in hand gesture recognition [9].

Using a combination of features leads to the problem of how to quantify their reliability. Ideally, the importance of each feature should be adapted to the changes in target pose and the surrounding background. This adaptation would improve the performance under changes not modeled by the tracker itself, hence removing the need for human intervention to re-tune the algorithm.

B. Contribution

We propose a multi-feature tracker that adaptively weights in a particle filtering framework the reliability of each feature. In the specific implementation, the target representation is based on color and orientation histograms. Moreover, we propose a feature uncertainty measure based on the determinant of a weighted covariance matrix of the target state, as sampled by a particle filter. Unlike traditional approaches, the uncertainty is based on the variability of the likelihood in the state space, and not on the likelihood itself. We modify the standard resampling strategy of the particle filter to incorporate the proposed estimate of the reliability. In addition to the above, we

E. Maggio and A. Cavallaro are with the Multimedia and Vision Group, Queen Mary University of London, Mile End Road, London E1 4NS (UK) (Tel: +44 20 7882 5165 Fax: +44 20 7882 7997) e-mail: {emilio.maggio, andrea.cavallaro}@elec.qmul.ac.uk. The authors acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC), under grant EP/D033772/1.

F. Smeraldi is with the Computer Vision group, Queen Mary University of London, Mile End Road, London E1 4NS (UK) (Tel: +44 20 7882 5167 Fax: +44 20 8980 6533) e-mail: fabri@dcs.qmul.ac.uk.

augment the color-based tracker with orientation histograms. Our orientation histogram representation is obtained from the eigenvalues of the structure tensor, providing a least-square estimate of the gradient which increases robustness to noise. Finally, we use the state parameters of the particles to normalize the orientation of the gradient and to select the scale of the derivative filters approximated by a scale-space approach, thus achieving rotation and scale invariance.

The paper is organized as follows. Section II discusses the previous work on feature integration for target tracking. Section III describes the selected features for target representation. Particle filtering and adaptive feature integration are discussed in Sec. IV. Section V presents the experimental results and the performance evaluation. Finally, Sec. VI concludes the paper.

II. Previous work

THE combination of multiple features to define the target model provides a higher degree of descriptiveness in video-based tracking. The integration can be performed at the *tracker level* or at the *measurement level*. While *tracker level* fusion allows the use of a wider range of trackers, fusion at the *measurement level* is preferable to avoid multiple runs of the full tracker, hence reducing the number of similar and redundant tracking hypotheses. A summary of multi-feature tracking algorithms is given in Tab. I.

Fusion at *tracker level* models each single-feature tracking algorithm as a black box. The problem is redefined by modeling the interaction between the outputs of the black boxes. An example of *tracker level* fusion is the use of multiple independent condensation algorithms for each feature, followed by the integration of the target estimations by multiplying the posterior probabilities [10]. If each feature spans a separate sub-space of the target state, a similar framework can also account for conditional feature dependencies [11]. The outputs of independent algorithms tracking localized parts of the target can be used as the observables of a Markov network [12]. A state variable is associated to each part; assuming linear and Gaussian interactions the compatibility between the states (i.e., the position of the parts) is modeled by the network. An algebraic criterion assesses the inter-part consistency, thus allowing the removal of inconsistent measurements. An alternative is to perform the fusion sequentially, considering the features as if they were available at subsequent time instants. The results of a blob detector, a color-based mean shift tracker, and a feature point tracker are incrementally incorporated by extended Kalman filtering [13]. The frame-by-frame measurement noise used by the filter for each feature acts as a feature reliability estimator. The measurement noise can also be estimated in a training phase [14], thus avoiding to adapt the feature contribution over time, but reducing the flexibility of the tracker under changing scene conditions.

When fusing multiple features at *measurement level*, single tracking algorithms combine the measurements internally. The phase coefficients of the wavelet decomposition can be considered as multiple features forming a time evolving template [15]. Each phase coefficient is modeled independently by a mixture of three components: a stable component, a fast varying component and a component that models outliers. The fusion is performed by the search procedure that gives more importance to stable coefficients. As all the measurements are generated with the same technique, they also present the same failure modalities [15]. A Markov model can be used to eliminate the measurements generated by clutter and to replace occluded measurements [16]. Also, the saliency maps of multiple features can be adaptively integrated as a weighted average, where the weights depend on the correlation between the saliency of each feature and the overall result [17]. However, this solution is limited to single target tracking (since the decision is based on the evaluation of the different descriptors on the whole frame), and it is valid only when consensus between the individual features is predominant [19]. To overcome this limitation, a particle filter framework can be used, thus evaluating the features only on the tracking hypotheses (particles) propagated by the algorithm. For example, multiple multi-modal features can be fused non-adaptively in a particle filter assuming conditional independence of the cues given the state [18]. Also, a particle filter followed by clustering can be used to discover multiple target positions [19]. However, the feature contribution is held constant and the adaptivity relies on the resampling step that discards particles with low likelihood. The contribution of each feature can be taken into account by multiplying likelihoods (assuming inter-feature independence), and by selecting the weights based on the distance between the tracking result of each feature and the global tracking result. Each weight is used as exponent of the corresponding likelihood [20]. This solution is equivalent to creating a weighted log-likelihood mixture [27]. A similar reliability measure is used in a voting framework to fuse five features for visual servoing [21]. Also a Bayesian network can model the dependency of multiple reliability scores to evaluate different features [22]. This method requires a training phase to learn the parameters of the network. To account for cooperative feature interaction, mutual information is used to quantify inter-feature agreement [23], and to assess feature reliability [24]. Feature interaction can be learned using a graphical model approximated by variational inference and Monte Carlo sampling [25]. Using color and shape information, the color state is iteratively updated by sampling the shape prior; whereas the shape state is updated by sampling the color prior. A graphical model coupled with inter-feature belief propagation has also been used to integrate color, shape and intensity [26]. However, as the final output is a set of three different states (each one associated with a feature), the fusion problem is only partially solved.

III. Features for target representation

LET us represent the candidate target area with an ellipse centered in $\mathbf{y} = (x, y)$, with length of the major axis h, eccentricity e, and rotation θ. These parameters define the state of the target $\mathbf{x}_t$, at time t, as

$$\mathbf{x}_t = (\mathbf{y}_t, h_t, e_t, \theta_t). \tag{1}$$

We describe the target area with two feature vectors. The first vector encodes the color properties of the target, while the

TABLE I

TRACKING ALGORITHMS COMBINING MULTIPLE FEATURES. TL: FUSION AT THE TRACKER LEVEL; ML: FUSION AT THE MEASUREMENT LEVEL.

	Ref.	Algorithm	Model features	Feature combination
TL	[10]	Condensation, Kalman filter	Template, blob, color	Non-adaptive product
	[11]	Particle filter	Color, contour	Product of conditionally dependent densities
	[12]	Kanade-Lucas, Particle Filter	Template	Bayesian network
	[13]	Extended Kalman filter	Blob, color, geometry	Sequential integration
	[14]	Condensation	Template, color	Covariance estimation
ML	[7]	Full search on motion predicted region	Color histogram, edge map	Non-adaptive linear combination
	[8]	Trust region search	Color histogram, edge density	Non-adaptive linear combination
	[15]	EM on affine parameters	Phase of wavelet coefficients	Higher contribution from stable coefficients
	[16]	Monte Carlo	Edge feature points on the contour	Clutter and occlusion modeling
	[17]	Saliency map fusion (full search)	Motion, color, position prediction, shape, contrast	Adaptive democratic integration
	[18]	Particle filter	Color, motion, sound	Non-adaptive likelihood factorization
	[19]	Multi-target clustered particle filter	Motion, color, Kalman prediction, shape, contrast	Non-adaptive linear combination
	[20]	Particle filter	Color, shape	Adaptive log-likelihood mixture
	[21]	Optimized search on window	Edge, disparity, color, template, motion	Adaptive voting
	[22]	Kalman filter	Color, motion, blob	Bayesian network
	[23]	Full search	Shape, color, template	Inter-feature mutual information
	[24]	Multiple hypothesis	Intensity, texture, color	Intra-feature mutual information
	[25]	Monte Carlo	Color, shape	Co-inference learning
	[26]	Monte Carlo	Color, shape, intensity change	Inter-feature belief propagation

second vector provides an invariant representation of the object shape and internal edges.

A. Color histograms

We code the color information of the state $\mathbf{x}$ (for clarity we drop the time subscript t) using part-wise histograms that incorporate both global and local target information in a single model [5]. Given a set $S_c = \{s_{c,j}\}_{j=1}^{N_{c,r}}$ of $N_{c,r}$ semi-overlapping parts of the target candidate ellipse, the part-wise normalized color histogram $f_c(\mathbf{x}) = \{f_c^{(u)}(\mathbf{x})\}_{u=1}^{N_{c,r} \cdot N_{c,b}}$, with $N_{c,b}$ color bins per part, is formed by concatenating the color histograms of each part normalized to one and then re-normalized again by multiplying by $1/N_{c,r}$. We use an elliptic kernel that lowers the contribution of the pixels that are closer to the border of the target. The first part $s_{c,1}$ is the whole target. To increase the sensitivity to rotations, four parts ($s_{c,2}$, $s_{c,3}$, $s_{c,4}$, and $s_{c,5}$) are obtained from the partition created by the two axes of the ellipse. Finally, to increase the sensitivity to scale changes, the inner and outer area of a concentric ellipse with same eccentricity, and half the axis size of the whole ellipse are considered ($s_{c,6}$ and $s_{c,7}$). By encoding the local distribution of the colors, this subdivision avoids representation ambiguities when the object is close to circular, and the ambiguity is now restricted to the case of circular objects with circular symmetry of the colors.

The similarity between a candidate and the model is defined as the distance $d(.)$ between the histogram associated with a candidate $f_c(\mathbf{x})$ and the model q_c, based on the Bhattacharyya coefficient [3],

$$d\left(f_c(\mathbf{x}), q_c\right) = \sqrt{1 - \sum_{u=1}^{N_{c,r} \cdot N_{c,b}} \sqrt{f_c^{(u)}(\mathbf{x}) \cdot q_c^{(u)}}}. \qquad (2)$$

Similarity measures like Eq. (2) can lead to unreliable candidate-model matches. Figure 1 shows an example of a lost track due to the use of color histograms only: the tracker is uncertain about the position of the target as the box on the bottom-left area of the image is a good candidate target region when the head is turning. To improve the target model distinctiveness additional information is needed.

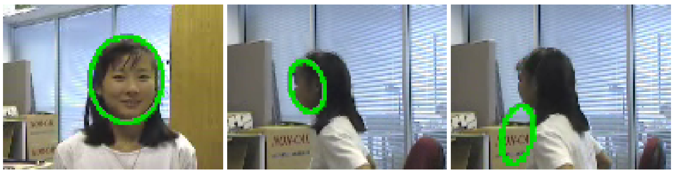


Fig. 1. Example of lost track using color histograms only. The color distribution of the box on the bottom-left of the image is similar to the color distribution of the face, thus misleading the tracker while the head is turning.

B. Orientation histograms

Orientation information is a desirable complement to color information in case of changes in illumination and background clutter. We exploit this information using a part-wise orientation histogram approximating the distribution of the gradient.

For each pixel $\mathbf{w}_{j,i}$ in the region of interest j, the magnitude of the gradient, $|\nabla I(\mathbf{w}_{j,i})|$, is accumulated on the bin corresponding to its orientation $\psi(\mathbf{w}_{j,i})$. A part-wise orientation histogram $f_o(\mathbf{x})$ is calculated, where the parts are the four sectors of the ellipse. To account for half-bin wide target rotations and spatial discontinuities, we use tri-linear interpolation to smooth the estimated histogram [28].

To increase *robustness to noise*, the gradient is evaluated using a least square estimate obtained from the structure tensor [29]

$$J(\mathbf{w}) = \int \rho(\mathbf{w} - \mathbf{w}') \left(\nabla I(\mathbf{w}')^T \nabla I(\mathbf{w}')\right) d\mathbf{w}', \qquad (3)$$

where $\rho(.)$ is a Gaussian kernel smoothing the estimate at the pixel position $\mathbf{w}$, and $J(.)$ is a 2×2 symmetric matrix. The best local fit to the direction of the gradient is the eigenvector $\mathbf{k}_{max}(\mathbf{w})$ of $J(\mathbf{w})$ associated with the largest eigenvalue $\lambda_{max}(\mathbf{w})$. The two eigenvalues $\lambda_{max}(\mathbf{w})$ and $\lambda_{min}(\mathbf{w})$ carry information about the local neighborhood: $\lambda_{min}(\mathbf{w}) \approx 0$ in the presence of a clear edge, while $\lambda_{max}(\mathbf{w}) \approx \lambda_{min}(\mathbf{w})$ if no single orientation predominates. A measure of *edge certainty* $G(.)$ can therefore be defined as

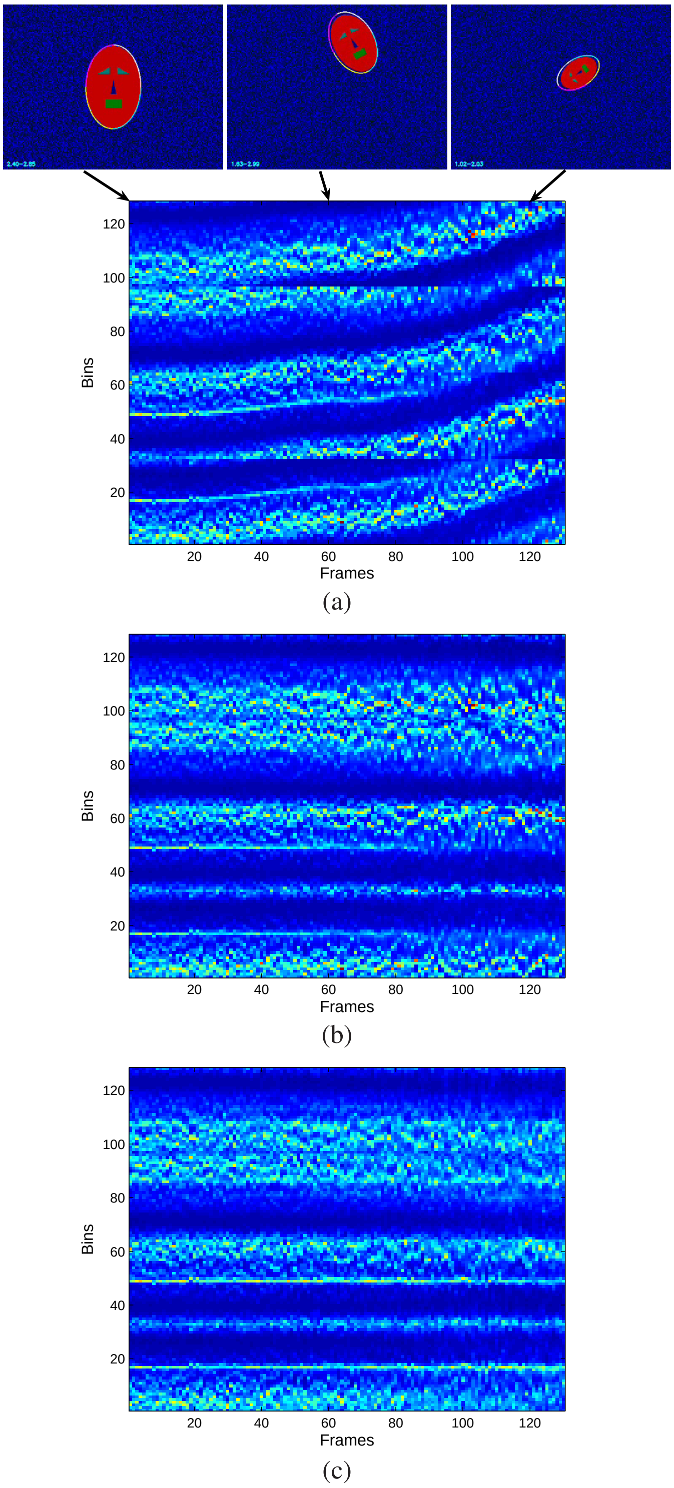
$$G(\mathbf{w}) = \sqrt[4]{\lambda_{max}(\mathbf{w})^2 - \lambda_{min}(\mathbf{w})^2}. \tag{4}$$

$G(.)$ highlights neighborhoods corresponding to strong straight edges, and penalizes neighborhoods with $\lambda_{min}(\mathbf{w}) \neq 0$ [30].

The orientation histogram based on the structure tensor is obtained by quantizing the range $[-\pi/2, \pi/2]$ into $N_{o,b}$ bins. For each position $\mathbf{w}$, the value of $G(\mathbf{w})$ is cumulated in the bin corresponding to the orientation $\phi(\mathbf{w})$ of the vector $\mathbf{k}_{max}(\mathbf{w})$. To account for targets moving through regions with different background intensity, vectors with opposite directions are cumulated onto the same bin. Although some information contained in the internal edges is discarded, a higher invariance level is achieved. To include only strong edges, a threshold is applied to $G(.)$. This threshold is set at the 10^{th} lowest percentile of the distribution of $G(.)$, as estimated from the target region in the previous frame. Figure 2 (a) illustrates with a toy target the properties of the orientation histograms: the orientation histograms are not invariant to target rotations, thus causing a shift of the peaks in the histogram.

Invariance to rotations has been previously addressed by blurring the histogram with a kernel [9], or by normalizing the gradient direction with respect to the dominant orientation [28]. However, if a kernel is used, the invariance is bounded by the kernel width, and a large kernel results in an excessive loss of information. Also, if the histogram presents multiple peaks with similar magnitude, the dominant orientation has to be chosen arbitrarily. To overcome these problems, we use the estimate θ of the target orientation provided by its state vector $\mathbf{x}$ (as sampled by the particle filter algorithm, see Sec. IV) [31]. The orientation is modeled by the rotation of the ellipse bounding the target area. The main idea is to shift the coefficients of the histogram according to θ. Note that the alignment is *not* based on the dominant orientation, but on the tracking hypotheses; these hypotheses are coherent with the target states at the previous time steps, thus helping to overcome degenerate situations.

Figure 2 (b) shows the rotation invariant orientation histograms. The peaks are now stabilized by the phase shifting mechanism. These orientation histograms share the scale invariance properties of normalized histograms. However, a problem arises under scale changes comparable with the original target size when computing the structure tensor. The derivative filters used in the computation of the gradient, and the smoothing kernel $\rho(.)$ of the structure tensor (Eq. (3)), have a scale parameter that determines the level of detail. For the representation to be truly invariant, the scale parameter should be adapted to the varying dimensions of each target candidate. If the scale of the filters is fixed in the first frame (Fig. 2 (b)), some prominent peaks are smoothed thus loosing important details of the representation. We achieve scale adaptation by convolving the original image with Gaussian derivative filters with different standard deviations σ_i, thus generating a derivative scale space. The orientation histogram of an ellipse with major axis h is then computed using the scale space related level $\sigma \approx h/r$, where r is a constant that determines the level of detail. Finally, Fig. 2 (c) shows that the proposed representation succeeds in preserving over time the main structures of the model histogram.



(a)

(b)

(c)

Fig. 2. Time evolution of different orientation histograms for the moving face showed in the first row (sample frames: 1, 60, and 120). (a) Rotation and scale variant histogram: the peaks shift over time. (b) Rotation invariant but scale variant histogram: the peaks are smoothed by large kernel values. (c) Scale and rotation invariant histogram (proposed representation): the values of the histogram are less affected by rotations and scale changes of the target.

IV. Multi-Feature Adaptive Particle Filter

IN this section we present a general procedure for the adaptive combination of multiple target representations in a single particle filter framework. Furthermore, we propose a strategy for the estimation of the feature contribution in the resampling procedure. Finally, we compare different reliability measures that estimate the relative importance of multiple features.

A. Feature combination with particle filtering

Particle filtering [32] solves the tracking problem by estimating the sequence of states $\mathbf{x}_t$, defined in Eq. (1), based on previous and current observations $\mathbf{z}_{1:t}$ (the image pixels observed up to time t). In a Bayesian approach, the problem consists of calculating the conditional density $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ (posterior). The posterior probability $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ of the target state is approximated with a sum of N_s Dirac functions (the *particles*) centered in $\left\{\mathbf{x}_t^i\right\}_{i=1}^{N_s}$ as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \approx \sum_{i=1}^{N_s} \omega_t^i \delta\left(\mathbf{x}_t - \mathbf{x}_t^i\right), \tag{5}$$

where ω_t^i are the weights associated with the particles that are calculated as

$$\omega_t^i \propto \hat{\omega}_{t-1}^i \frac{p(\mathbf{z}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{z}_t)}. \tag{6}$$

Here $q(.)$ is the proposal distribution used to sample the particles, and $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the state transition model, which is used to propagate the particles toward new states, i.e., $q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{z}_t) \propto p(\mathbf{x}_t|\mathbf{x}_{t-1})$. To discard particles with lower weights, a resampling step is applied before propagation. $\hat{\omega}_{t-1}^i$ is the particle weight after the resampling step that draws the $\left\{\mathbf{x}_t^i\right\}_{i=1}^{N_s}$ from the set $\left\{\mathbf{x}_{t-1}^i\right\}_{i=1}^{N_s}$ according to the resampling function $\left\{a_{t-1}^i\right\}_{i=1}^{N_s}$ [33]. The resampling function defines the probability of each particle $\mathbf{x}_{t-1}^i$ to generate a new sample at time t. This leads to $\hat{\omega}_{t-1}^i = \omega_{t-1}^i/(N_s \cdot a_{t-1}^i)$. For arbitrary resampling functions

$$\omega_t^i \propto \frac{\omega_{t-1}^i}{a_{t-1}^i} p(\mathbf{z}_t|\mathbf{x}_t^i). \tag{7}$$

Usually $a_{t-1}^i = \omega_{t-1}^i$ (i.e., $\hat{\omega}_{t-1}^i = 1/N_s \ \forall i$), hence

$$\omega_t^i \propto p(\mathbf{z}_t|\mathbf{x}_t^i) \tag{8}$$

that is, the weights are proportional to the likelihood of the observation vector.

In the multi-feature case, the likelihood $p(\mathbf{z}_t|\mathbf{x})$ should be dependent on the distance from the model calculated for each feature. Suppose that for each feature m ($m = 1, \ldots, M$) we can evaluate the likelihood $p_m(\mathbf{z}_t|\mathbf{x}_t)$. Hence

$$\{p_m(\mathbf{z}_t|\mathbf{x}_t)\}_{m=1}^M \tag{9}$$

is known for all t (in the specific implementation for this paper $M = 2$). The overall likelihood is generated by linear combination of the single features [19] as the likelihood mixture

$$p(\mathbf{z}_t|\mathbf{x}) = \sum_{m=1}^{M} \alpha_{m,t} p_m(\mathbf{z}_t|\mathbf{x}_t), \tag{10}$$

where $\alpha_{m,t}$ is a mixture coefficient, and

$$\sum_{m=1}^{M} \alpha_{m,t} = 1. \tag{11}$$

In the case of object classification, the sum rule was demonstrated to outperform the product rule and other classifier combinations schemes by being less sensitive to ambiguous and inconsistent measurements. We also argue that similarly in object tracking the sum rule is less sensitive to clutter and targets with similar appearance. Moreover, this strategy is in line with the assumption that humans perceive visual content through a sum of multiple features weighted by their reliability [34].

We calculate the likelihood of each feature using the distance from the model histograms, defined in Eq. (2) as

$$p_m(\mathbf{z}_t|\mathbf{x}) = e^{-\left(\frac{d(f_m(\mathbf{x}), q_m)}{\sigma}\right)^2}. \tag{12}$$

The histogram $f_m(\mathbf{x})$ defined by the state $\mathbf{x}$ is calculated over the pixels of the observation vector (the image) $\mathbf{z}_t$. The exponent is used to obtain a smooth likelihood thus facilitating the final state estimation. The value of σ, which models the noise on the measurements, is determined experimentally based on the fact that the gradient orientation is more affected by noise than color and that the finer the quantization, the higher is the impact of the noise.

The best state at the time t is derived based on the discrete approximation created by the weighted particles. The most common solution is the Monte Carlo approximation of the expectation $\mathbb{E}(\mathbf{x}_t|\mathbf{z}_{1:t})$ calculated as the weighted average of the particles $\mathbf{x}_t^i$.

B. Multi-feature resampling

When using the resampling function $a_{t-1}^i = \omega_{t-1}^i$ in Eq. (7), for the multi-feature case, the particles are drawn proportionally to the mixed likelihood weights of Eq. (10). When the algorithm degenerates (i.e., all but one feature give negligible contribution), most particles are resampled from a single-feature ignoring the other components of the mixed likelihood. As the evaluation of the reliability of each feature requires a set of particles that accurately represents all the components of the mixture as defined in Eq. (10), we introduce a multi-feature resampling strategy. The resampling function is defined as

$$a_t^i = \sum_{m=1}^{M} \beta_{m,t} p_m(\mathbf{z}_t|\mathbf{x}_t^i) \quad i = 1, ..., N_s, \tag{13}$$

where

$$\beta_{m,t} = \begin{cases} \alpha_{m,t} & \text{if } \alpha_{m,t} > T \\ T & \text{otherwise} \end{cases} \quad m = 1, ...M, \tag{14}$$

and T defines the lower bound for the number of particles resampled from each feature. After thresholding, we normalize the $\{\beta_{m,t}\}_{m=1}^M$ to one. We will refer to the multi-feature particle filter with the proposed resampling procedure as MF-PFR and to the multi-feature particle filter with standard resampling (see Eq. (8)) as MF-PF. When the weights are updated on line, we obtain the Adaptive Multi-Feature Particle Filter (AMF-PF) that is described in Algorithm 1.

Algorithm 1 Multi-Feature Adaptive Particle Filter

```
$\left[\{\mathbf{x}^i_{t-1}, \omega^i_{t-1}\}^{N_s}_{i=1}, \{\alpha_{m,t-1}\}^M_{m=1}\right] \rightarrow \left[\{\mathbf{x}^i_t, \omega^i_t\}^{N_s}_{i=1}, \{\alpha_{m,t}\}^M_{m=1}\right]$
Compute $\{a^i_{t-1}\}^{N_s}_{i=1}$ according to Eq. (13)
Resample the particles from $\{\mathbf{x}^i_{t-1}, a^i_{t-1}\}^{N_s}_{i=1}$
for $i = 1 : N_s$ do
    Draw $\mathbf{x}^i_t$ from $p(\mathbf{x}_t|\mathbf{x}_{t-1})$
    Compute $\{p_m(\mathbf{z}_t|\mathbf{x}^i_t)\}^M_{m=1}$ according to Eq. (12)
end for
Compute $\{\alpha_{m,t}\}^M_{m=1}$
for $i = 1 : N_s$ do
    Compute $p(\mathbf{z}_t|\mathbf{x}^i_t)$ according to Eq. (10)
    Assign the particle a weight $\omega^i_t$ according to Eq. (7)
end for
```

C. Feature weighting

To compute the likelihood as in Eq. (10), we estimate the mixture coefficients $\alpha_{m,t}$ based on each feature reliability.

1) Existing reliability measures: The reliability of a feature can be computed based on the average value of the saliency over the whole frame [17] (Note that as the framework in [17] is not probabilistic, the term saliency instead of likelihood is used). Due to the discrete nature of the particle filter approximation, we evaluate the saliency on the states defined by the particles, and not on the whole frame. The measure $\gamma^1_{m,t}$ (*distance to average*) for feature m is defined as

$$\gamma^1_{m,t} = \mathcal{R}\left(p_m(\mathbf{z}_t|\hat{\mathbf{x}}_t) - \langle p_m(\mathbf{z}_t|\mathbf{x}_t)\rangle\right), \tag{15}$$

where $\hat{\mathbf{x}}_t$ is the state determined by the particle with maximum fused likelihood, defined as

$$\hat{\mathbf{x}}_t = \arg\max_i \left\{p(\mathbf{z}_t|\mathbf{x}^i)\right\}, \tag{16}$$

and $\langle p(\mathbf{z}_{m,t}|\mathbf{x}_t)\rangle$ is the average likelihood of feature m over the set of particles. $\mathcal{R}(.)$ is the ramp function

$$\mathcal{R}(x) = \left\{\begin{array}{ll} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{array}\right. . \tag{17}$$

An alternative solution, here referred to as $\gamma^2_{m,t}$, substitutes $\hat{\mathbf{x}}_t$ with the best particle selected separately by each feature:

$$\hat{\mathbf{x}}_{m,t} = \arg\max_i \left\{p_m(\mathbf{z}_t|\mathbf{x}^i_t)\right\}. \tag{18}$$

Feature reliability can also be estimated based on the level of agreement between each feature and the overall tracker result [20]. The contribution of each feature is a function of the Euclidean distance, $\bar{E}_{m,t}$, between the center of the best state estimated by feature m and the center of the state obtained combining the features using the reliability scores of time $t-1$. The corresponding reliability score, $\gamma^3_{m,t}$ (*centroid distance*), is computed by smoothing $\bar{E}_{m,t}$ with a sigmoid function

$$\gamma^3_{m,t} = \frac{\tanh(-a\bar{E}_{m,t} + b) + 1}{2}, \tag{19}$$

where a, b are constants. $a = 0.4$ pixels^{-1} and $b = 3$ are the values used in the original paper [20], and will be used for the evaluation in Sec. V-C. Note that this measure does not include any information about the error in the estimation of the size and the rotation of the target.

2) Proposed reliability measure: We weight the influence of each feature based on their spatial uncertainty [35]. We propose to estimate the spatial uncertainty analyzing the eigenvalues of the covariance matrix $C_{m,t}$ of the particles $\mathbf{x}^i_t$ weighted by the likelihood, and computed for each feature m at time t. Although the state space we use is 5-dimensional, for illustrative purposes we now define $C_{m,t}$ for a 2-dimensional state, $\mathbf{x} = (u, v)$. Then the 2×2 normalized covariance matrix is

$$C_m = \left[\begin{array}{cc} \frac{\sum_{i=1}^{N_s} l_m(u^i,v^i)(u^i-\hat{u})^2}{\sum_{i=1}^{N_s} l_m(u^i,v^i)} & \frac{\sum_{i=1}^{N_s} l_m(u^i,v^i)(u^i-\hat{u})(v^i-\hat{v})}{\sum_{i=1}^{N_s} l_m(u^i,v^i)} \\ \frac{\sum_{i=1}^{N_s} l_m(u^i,v^i)(u^i-\hat{u})(v^i-\hat{v})}{\sum_{i=1}^{N_s} l_m(u^i,v^i)} & \frac{\sum_{i=1}^{N_s} l_m(u^i,v^i)(v^i-\hat{v})^2}{\sum_{i=1}^{N_s} l_m(u^i,v^i)} \end{array}\right]. \tag{20}$$

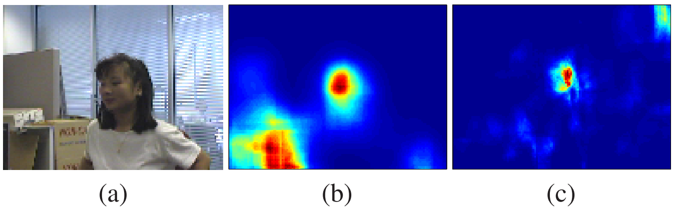
For a more readable notation we have omitted t, and used $l_m(\mathbf{x}_t)$ instead of $p_m(\mathbf{z}_t|\mathbf{x}_t)$. The extension to the 5-D case is straightforward and a 5×5 covariance matrix is obtained. We can now define the uncertainty $U_{m,t}$ as

$$U_{m,t} = \sqrt[D]{\prod_{k=0}^{D} \lambda^{(k)}_{m,t}} = \sqrt[D]{\det(C_{m,t})}, \tag{21}$$

which is related to the volume of the hyper-ellipse having the eigenvalues $\{\lambda^{(k)}_{m,t}\}^D_{k=1}$ as semi-axes. D is the dimensionality of the state space. The determinant $\det(.)$ is used instead of the sum of the eigenvalues to avoid problems related to state dimensions with different ranges (i.e., position versus size or orientation). The larger the hyper-volume, the larger is the uncertainty of the corresponding feature about the state of the target. The corresponding reliability score (*spatial uncertainty*) is defined as

$$\gamma^4_{m,t} = 1/U_{m,t}. \tag{22}$$

The importance of each feature is therefore the reciprocal of its uncertainty. We compare two different versions of this score. The first version, $\gamma^4_{m,t}$, computes the average $(\hat{u}, \hat{v})$ for Eq. (21) from the particle states weighted by the fused likelihood using the reliability estimated at time $t-1$. This measures the likelihood spread compared with the tracker result. The second version, $\gamma^5_{m,t}$ uses the weights of each feature to compute the average $(\hat{u}_m, \hat{v}_m)$ which is substituted for $(\hat{u}, \hat{v})$ in Eq. (20), thus measuring the internal spread of each likelihood (Eq. (9)). Figure 3 shows a typical situation where measuring the spatial uncertainty of a feature helps the tracker. While tracking the face the color histograms information is ambiguous, as the box on the bottom-left has a similar color to the skin. On the other hand, orientation information is more discriminative (less spatially spread) on the real target. Moreover, the reliability estimation based on particle filter sampling allows us to assign different reliability scores to different targets in the scene. When the targets are far in state space, Eq. (21) determines the discriminative power of a feature in separating the target from the background. When the targets are close, the two sets of hypotheses will overlap. Hence, due to the multi-modality of the likelihoods, the uncertainty defined by Eq. (21) will increase.



(a) (b) (c)

Fig. 3. Comparison between the model-candidate likelihood of color and orientation histograms in a face tracking scenario. The target model is computed on the corresponding frame of Fig. 5. (a) Frame under analysis. (b) Spatial spread of the color likelihood. (c) Spatial spread of the orientation likelihood. A reliability score measuring the spatial spread could improve the tracker performance by assigning to the orientation a larger weight than that of the color.

To smooth temporal variations, the reliability scores, $\gamma^j_{m,t}$, undergo temporal filtering using the leaky integrator

$$\alpha^j_{m,t} = \tau\alpha^j_{m,t-1} + (1-\tau)\gamma^j_{m,t}, \tag{23}$$

where $\tau \in [0,1]$ is the forgetting factor. The lower τ, the faster is the update of $\alpha^j_{m,t}$. To satisfy Eq. (11), it is sufficient to enforce the condition $\sum_{m=1}^{M} \gamma^j_{m,t} = 1$.

Figure 4 compares the time evolution of the reliability scores while a head undergoes a 360^o out-of-plane rotation. When the head starts rotating, the contribution of the gradient should increase as the color distribution changes significantly. When the target is again in a frontal pose, the gradient contribution should decrease and approach its initial value. The score α^1 has a high variability, caused by the ramp function of Eq. (17). The likelihood evaluated in the best combined state is often lower than the average likelihood, thus resulting in $\gamma^1_m = 0$ and a rapid variation of α^1. Unlike α^1, α^2 is not influenced by the ramp, since the likelihood is measured on the best particle of each feature separately. α^2 correctly increases the importance of the orientation histogram during the rotation. However, other variations are generated when no adaptation is expected. Similar considerations can be drawn for α^3: the high variability is not always motivated by real appearance changes. Before the head rotation, the two scores α^4 and α^5 behave similarly. However, only α^4 has an adaptation profile compatible with the head rotation. Section V-C will provide quantitative results to support these observations.

V. Experimental validation

A. Test conditions

We demonstrate the proposed tracker on a dataset composed of 12 heterogeneous targets extracted from 9 different tracking sequences (Tab. II). Four head targets (H1, H2, H3, and H4) are from a public dataset[1] and other two (H5, and H6) are part of an in-house dataset[2]. Four pedestrians (P1, P2, P3, and P4) are extracted from the PETS 2001 dataset; P5 is from the CAVIAR dataset[3]. Finally the target O1 is extracted from a sequence generated using an omni-directional camera. Figure 5 shows sample frames highlighting the targets.

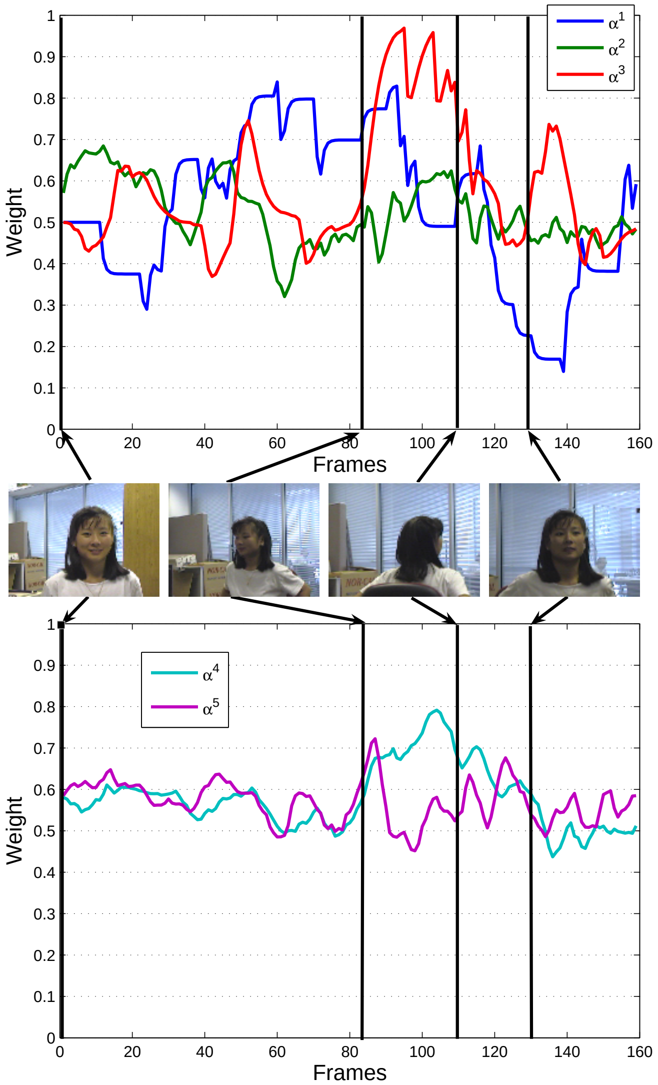


Fig. 4. Comparison of orientation histogram weights for the adaptive feature combination (α^1, α^2: distance to average; α^3: centroid distance; α^4, α^5: spatial uncertainty). The tracker is initialized as showed in the top-left image of Fig. 5.

The parameters of the tracker were set experimentally, and are the same for all the targets (the only exception is the standard deviation for H5, as described below). The color histograms are calculated in the RGB space with $N_{c,b} = 8 \times 8 \times 8$ bins, while the orientation histograms are calculated using $N_{o,b} = 32$ bins. As all the head targets perform unpredictable abrupt shifts, the particle filter uses a zero-order motion model $\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{n}_t$, where $\mathbf{n}_t$ is a multivariate Gaussian random variable with $\sigma_{h,t} = 0.05 \cdot h_{t-1}$, $\sigma_e = 0.021$, and $\sigma_\theta = 5^o$. $\sigma_x = \sigma_y = 5$ for all the targets except for H5 where $\sigma_x = \sigma_y = 14$. The values of σ in Eq. (12) are set to $\sigma_c = 0.09$ for the color, $\sigma_o = 0.13$ for the orientation. The particle filter uses 150 samples per frame. The time filtering parameter τ for adaptive tracking is $\tau = 0.75$, and a minimum of 45 particles is drawn from distribution of each single feature (i.e., $T = 0.3$).

[1] http://www.ces.clemson.edu/~stb/research/headtracker/seq

[2] The sequences and the ground truth are available at http://www.elec.qmul.ac.uk/staffinfo/andrea/multi-feature.html

[3] EC Funded CAVIAR project/IST 2001 37540, the dataset is available at http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

TABLE II
DESCRIPTION OF THE TRACKING DATASET.

Targets	Frame size	Frame rate	Characteristics
H1, H2, H3, H4	128 × 96	30fps	Scale changes, clutter, occlusions
H5	PAL	25fps	Clutter, self-occlusions
H6	320 × 240	12.5fps	Abrupt shifts, clutter, partial occlusions
P1, P2, P3, P4	PAL	25fps	Clutter, occlusions
P5	384 × 288	25fps	Clutter
O1	352 × 288	25fps	Omni-directional camera, occlusions

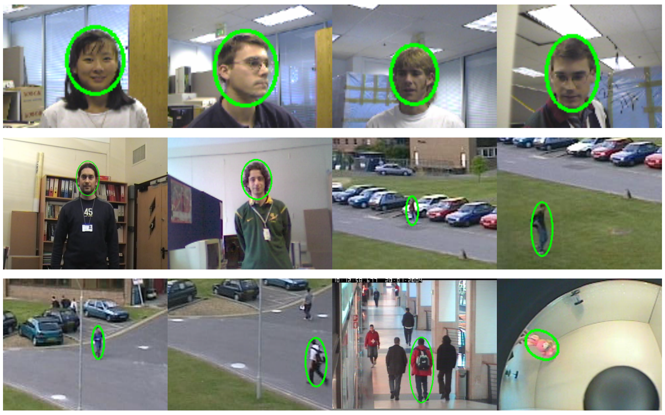


Fig. 5. The targets of the evaluation dataset. (From top-left to bottom-right) Head targets: *seq_mb* (H1), *seq_sb* (H2), *seq_jd* (H3), *seq_villains2* (H4), *Toni* (H5), *Emilio* (H6); pedestrians: (P1), (P2), (P3), (P4), and (P5); a toy bunny (O1).

B. Performance evaluation

To compare different trackers, we estimate the error of their results. The *error measure* quantifies the discrepancy between the target estimations and the manually generated ground-truth targets[3]. The ground-truth consists of the set of 5 parameters of the ellipse (centroid coordinates, size, eccentricity and rotation) best fitting the target at each frame. Let $t_p(t)$ be the number of true positive pixels (i.e., pixels belonging to both the ground-truth target and the estimated target) in each frame t. Let $|.|$ denote the cardinality of a set. If $|A_g(t)|$ and $|A_e(t)|$ are the ground-truth and the estimated target area respectively, then the error of the estimation $\mathcal{P}(t)$ at time t can be defined as

$$\mathcal{P}(t) = 1 - \frac{2t_p(t)}{|A_e(t)| + |A_g(t)|}. \tag{24}$$

This performance measure rewards candidates with a high percentage of true positive pixels, and with few false positives and false negatives avoiding the asymmetry problem of other area based measures [36]. Furthermore, unlike centroid based measures, $\mathcal{P}(.)$ accounts also for errors in the estimation of size, and eccentricity, and in case of lost detection is not dependent on the position where the tracker is stuck (i.e., $\mathcal{P}(.)$ saturates to 1). The quality measure of a whole track is obtained by averaging $\mathcal{P}(t)$ over the frames where the target is visible. Since particle filter is a probabilistic algorithm, each tracker is run 20 times for each sequence of the dataset. The seed of the function generating the random Gaussian variable $\mathbf{n}$ for state transition model is initialized using the processor clock. For each target in a sequence we calculate the average, minimum and maximal error and its standard deviation over the runs: a good tracker is characterized not only by a small average error, but also by small variations in the error in different runs.

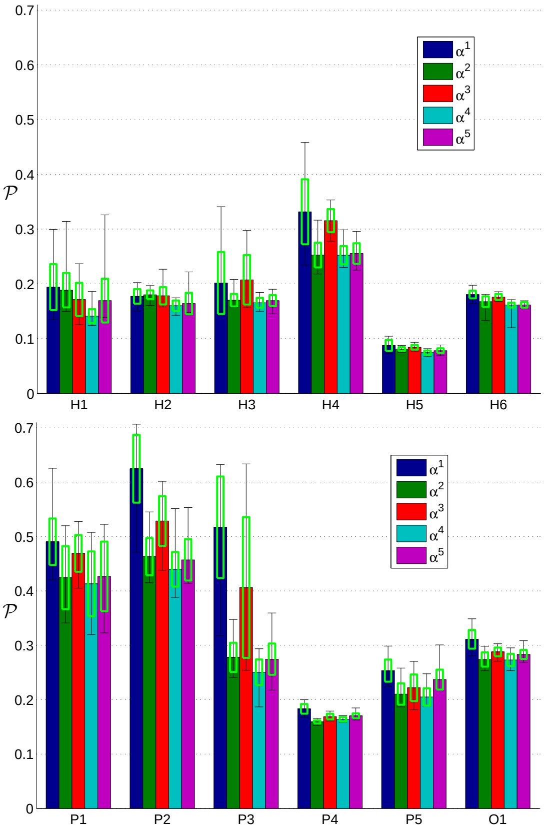


Fig. 6. Tracking results for different feature weighting strategies. The bars represent the average error, the boxes represent the standard deviation of the error, and the error bars represent the maximal and minimal error on each sequence.

C. Comparison of feature weighting strategies

Experimental results comparing the feature weights (Sec. IV-C) are shown in Fig. 6. The five scores achieve comparable performances on H5, H6 and P4, and present large error differences in sequences with occlusions and clutter (H3, H4, P1, P2, and P3). In particular α^1 and α^3 leads to poor results compared to α^2, α^4 and α^5 in H3, H4, P1, P2, and P3. This confirms that the two scores with faster variability are less accurate, especially in sequences with false targets and clutter. In fact, the more conservative score α^2 results in a more stable performance on the same targets. The score α^4 consistently yields to the lowest error and lowest standard deviation across different target typologies and tracking issues. In particular a large performance gap is achieved on H1, P2, and P3. This confirms the comments related to Fig. 4. The score α^5 is more accurate than α^1, α^2 and α^3, but performs worse than α^4. For this reason α^4 will be used as feature reliability score in Adaptive Multi-Feature Particle Filter (AMF-PFR) adopting the resampling procedure described in Sec. IV-B.

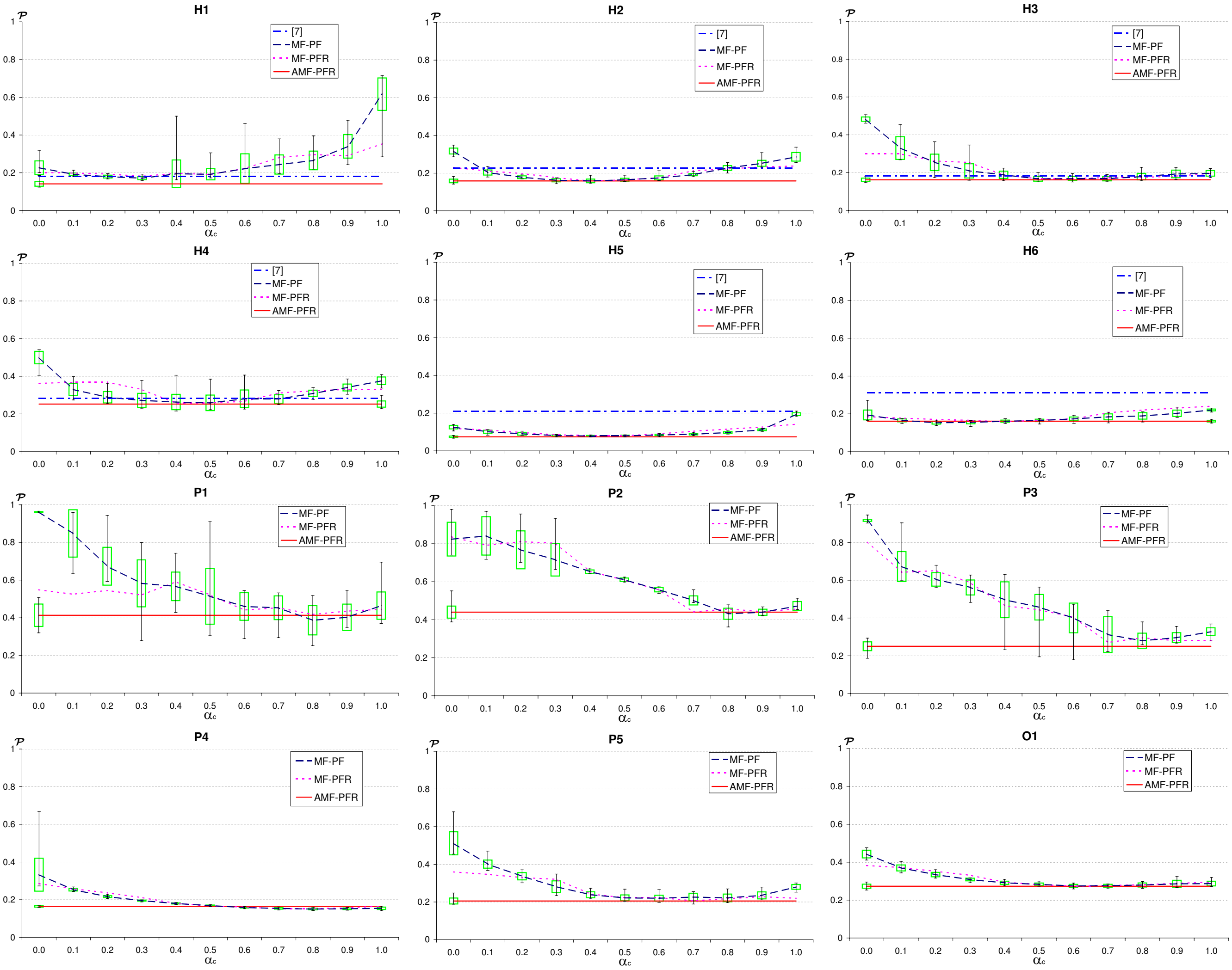


Fig. 7. Comparison of tracking results for the proposed algorithm (AMF-PFR), different fixed combinations of the features with (MF-PFR) and without (MF-PF) multi-feature resampling, and the head tracker proposed in [7]. The average distance from the ground truth (lines), the standard deviation (boxes) and the maximal and minimal errors (error bars) are plotted against the weight given to the color feature. For readability purpose the error bars and standard deviations of the non-adaptive algorithms are displayed only for MF-PF (comparable results are obtained with MF-PFR).

D. *Tracker comparison*

We first compare the proposed algorithm (AMF-PFR) against two trackers with various fixed weight combinations, namely MF-PFR (tracker with multi-feature re-sampling) and MF-PF (tracker without multi-feature re-sampling). Next, we compare AMF-PFR with the popular tracker proposed by Birchfield [7]. From now on we will refer to this tracker with its bibliography reference number, [7].

Figure 7 shows the performance comparison between the *adaptive* and the *non-adaptive* trackers. The results related to MF-PFR and MF-PF are obtained by fixing a priori the color importance α_c. Note the large performance improvements when moving from single feature (i.e., MF-PF with $\alpha_c = 0$ and $\alpha_c = 1$) to multi-feature algorithms. It is worthy noticing that the optimal working point, $\hat{\alpha}_c$, of the non-adaptive trackers (MF-PF, MF-PFR) varies from target to target. For example $\hat{\alpha}_c = 0.3$ in H1, while $\hat{\alpha}_c = 0.6$ in H3. In these two cases MF-PF and MF-PFR require manual tuning to achieve optimal performance, whereas in AMF-PFR the adaptation is automated. Furthermore the error of AMF-PFR is comparable with or lower than the best result of the non-adaptive algorithms (MF-PF and MF-PFR).

On H1 and H5 the error of AMF-PFR is 17% and 5% respectively lower than the error at the best working point of MF-PFR. Figure 9 shows sample frames on H1 from the run with the closest error to the average: MF-PF (first row) and AMF-PFR (second row). As the target changes scale (the first four columns of Fig. 9 show a 1/4 of octave scale change) the scale of the filter is adapted. The numbers superimposed on the images are the minimum and maximum of the standard deviation of the Gaussian derivative filters that are used to generate the scale space (Sec. III-B). When the head turns (second to fourth column) or a severe occlusion happens (second last column) the adaptive weighting reduces the tracking error. On H6, although AMF-PFR is more accurate than most of the fixed combinations of weights, its result is 5% worse than the best non-adaptive (manually set) result. In this sequence scale changes, illumination changes and a partial occlusion

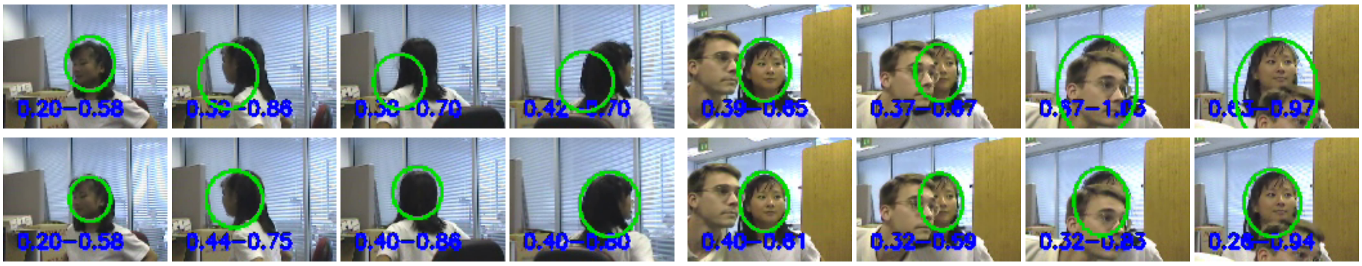


Fig. 9. Sample results from the *average run* on target H1 (frames 82, 92, 100, 109, 419, 429, 435, and 442). First row: non-adaptive multi-feature tracker (MF-PF). Second row: adaptive multi-feature tracker (AMF-PFR). When the target appearance changes AMF-PFR achieves reduced tracking error by varying the importance of the features over time.

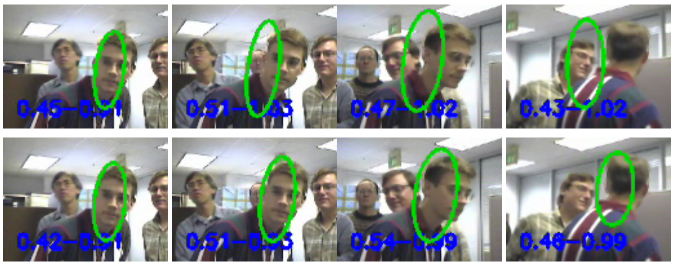


Fig. 8. Sample results from the *worst run* on target H4 (frames 116, 118, 126, and 136). First row: non-adaptive multi-feature tracking (MF-PF). The color contribution is fixed to 0.5, i.e. the value that gives the best average result on MF-PF. Second row: adaptive multi-feature tracker (AMF-PFR). Note that MF-PF is attracted by a false target, while the proposed method (*worst run*) is still on target.

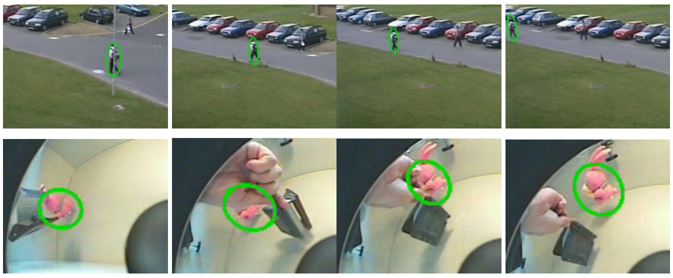


Fig. 10. Sample results of adaptive multi-feature tracker (AMF-PFR). First row: target P4 (frames 38, 181, 273, and 394). Second row: target O1 (frames 332, 368, 393, and 402).

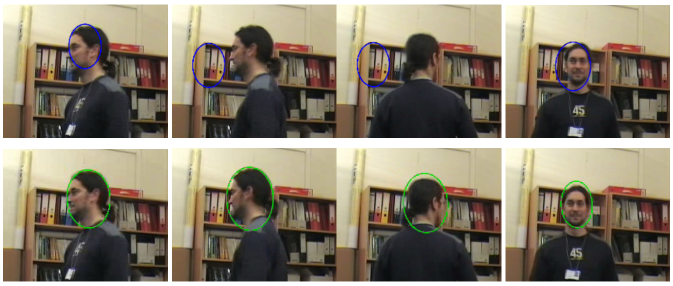


Fig. 11. Sample tracking results on target H5 (frames 200, 204, 218, and 247). First row: elliptic head tracker ([7]). Second row: adaptive multi-feature tracker (AMF-PFR). Unlike [7], the gradient information used in the AMF-PFR target model manages to separate the target from the clutter.

occur; both color and orientation models are unable to describe correctly the target, and this results in a sub-optimal adaptation of α. In this case, a larger pool of features could help to improve the effectiveness of the adaptation.

The results on P1, P2, P3, P4, and P5 (Fig. 7) show how the algorithm adapts when one feature is more informative. The edge distribution of a pedestrian has fast time variability; hence the orientation histogram does not contribute significantly to improve the tracker performance. By allowing time adaptation AMF-PFR manages to achieve a result that is similar to the best fixed combination of the features. Figure 10 (first row) shows sample results on P4. AMF-PFR successfully tracks the target despite the presence of clutter with similar colors (the white car), and despite the occlusion generated by the lamp post. Similarly for O1 (Fig. 10, second row): the occlusion generated by the hand is overcome thanks to the multiple hypothesis generated by PF and to the flexibility of the target representation.

Finally, the standard deviations and error bars of Fig. 7 show that the error is more stable for AMF-PFR than for the non-adaptive counterpart MF-PF. This is more evident in H1, H4, and P3, where a small variation of the weights results in MF-PF loosing the track. For example, Fig. 8 shows the results of the *worst run* in terms of error on target H4: MF-PF (first row) is attracted by false targets, and the track is lost during the rotation of the head. Although AMF-PFR (second row) does not accurately estimate the target size the object is continuously tracked.

Figure 7 shows also that AMF-PFR outperforms [7] on all the 6 head targets. Figure 11 shows how the cluttered edge responses generated by the bookshelf highly affect the performance of [7]. We believe that the performance improvement is due to representation of the gradient based on the orientation histograms. While [7] encodes only the information of the edges on the border of the object, the orientation histograms used in AMF-PFR represent also the distribution of the internal structures that are less likely affected by clutter.

On a Pentium 4 3GHz, a non-optimized implementation of AMF-PFR runs at 13.2fps on H1 (average area: 1540 pixels), 7.4fps on P4 (2794 pixels), and 2.2fps on H5 (10361 pixels). The computational complexity approximately grows linearly with the target area. It is also worth noticing that the complexity does not depend on the frame size, as the processing is done on a region of interest around the target. AMF-PFR spends 60.3% of the time computing the orientation histograms, 31.1% on the color histogram, 7.9% on the recursive propagation of the particles, and only 0.7% computing the feature reliability scores. The computational cost associated to the orientation histogram could be reduced by using an optimized implementation of the Gaussian scale-space [28].

VI. Conclusions and Future Work

We presented a multi-feature tracking algorithm that adaptively weights the contribution of each feature based on their reliability. A novel reliability score based on the weighted covariance matrix was proposed in a particle filter framework. The extension of this approach to particle filtering was not straightforward and is part of our contribution. The proposed adaptive particle filter algorithm improves the flexibility of the representation by exploiting the complementarities of the failure modes of the various descriptors in a simple and efficient way.

Experimental results over a set of real-world heterogeneous targets showed that the adaptive multi-feature representation formed by a combination of color and orientation histograms is more descriptive and leads to more accurate results than a single-feature representation, and outperforms or matches the optimal manually selected combination of the features. The proposed feature reliability score is general and can be extended to a larger set of features.

Future work includes the investigation of fusion mechanisms that account for inter-dependencies between features. Moreover, selection algorithms could be employed to dynamically disable redundant features to improve the computational efficiency of the algorithm. Also, we aim to investigate an integrated probabilistic treatment of the interaction between feature reliability estimation and particle filter sampling. Finally, we are studying a robust model update criterion driven by the estimates of the feature reliability in order to achieve longer-term tracking under varying conditions.

References

[1] M. Isard and J. MacCormick, "Bramble: A bayesian multiple-blob tracker." in *Proc. of International Conference on Computer Vision*, vol. 2, Vancouver, Canada, July 2001, pp. 34–41.

[2] A. Cavallaro, O. Steiger, and T. Ebrahimi, "Tracking video objects in cluttered background," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 15, no. 4, pp. 575–584, 2005.

[3] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[4] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. of the European Conference on Computer Vision*, vol. 1, Copenhagen, Denmark, May-June 2002, pp. 661–675.

[5] E. Maggio and A. Cavallaro, "Multi-part target representation for colour tracking," in *Proc. of IEEE International Conf. on Image Processing*, vol. 1, Genoa, Italy, Sept. 2005, pp. 729–732.

[6] S. Birchfield and S. Rangarajan, "Spatiograms versus histograms for region-based tracking," in *Proc. of IEEE Conf. on Comp. Vis. and Pattern Recog.*, vol. 2, San Diego, CA, USA, June 2005, pp. 1158–1163.

[7] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *Proc. of IEEE Conf. on Comp. Vis. and Pattern Recog.*, Santa Barbara, CA, USA, June 1998, pp. 232–237.

[8] T. Liu and H. Chen, "Real-time tracking using trust-region methods," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 3, pp. 397–402, 2004.

[9] W. Freeman and M. Roth, "Orientation histograms for hand gesture recognition," in *Proc. of Workshop on Autom. Face and Gesture Recognition*, Zurich, CH, June 1995, pp. 296–301.

[10] I. Leichter, M. Lindenbaum, and E. Rivlin, "A probabilistic framework for combining tracking algorithms," in *Proc. of IEEE Conf. on Comp. Vis. and Pattern Recog.*, vol. 2, Washington, DC, USA, June-July 2004, pp. 445–451.

[11] F. Moreno-Noguer, A. Sanfeliu, and D. Samaras, "Integration of conditionally dependent object features for robust figure/background segmentation," in *Proc. of International Conference on Computer Vision*, Washington, DC, USA, June 2005, pp. 1713–1720.

[12] G. Hua and Y. Wu, "Measurement integration under inconsistency for robust tracking," in *Proc. of IEEE Conf. on Comp. Vis. and Pattern Recog.*, New York, NY, USA, June 2006, pp. 650–657.

[13] H. Veeraraghavan, P. Schrater, and N. Papanikolopoulos, "Robust target detection and tracking through integration of motion, color, and geometry," *Comput. Vis. Image Underst.*, vol. 103, no. 2, pp. 121–138, 2006.

[14] J. Sherrah and S. Gong, "Fusion of perceptual cues using covariance estimation," in *Proc. of British Machine Vision Conference*, Nottingham, UK, Sept. 1999, pp. 564–573.

[15] A. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 10, pp. 1296–1311, 2003.

[16] Y. Wu, G. Hua, and T. Yu, "Switching observation models for contour tracking in clutter," in *Proc. of IEEE Conf. on Comp. Vis. and Pattern Recog.*, Madison, WI, USA, June 2003, pp. 295–304.

[17] J. Triesch and C. von der Malsburg, "Democratic integration: Self-organized integration of adaptive cues," *Neural Computation*, vol. 13, no. 9, pp. 2049–2074, 2001.

[18] P. Perez, J. Vermaak, and A. Blake, "Data fusion for visual tracking with particles," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 495–513, 2004.

[19] M. Spengler and B. Schiele, "Towards robust multi-cue integration for visual tracking," *Lecture Notes in Computer Science*, vol. 2095, pp. 93–106, 2001.

[20] C. Shen, A. van den Hengel, and A. Dick, "Probabilistic multiple cue integration for particle filter based tracking," in *7th International Conference on Digital Image Computing (DICTA'03)*, Sydney, AU, Dec. 2003, pp. 309–408.

[21] D. Kragic and H. Christensen, "Cue integration for visual servoing," *IEEE Trans. Robot. Automat.*, vol. 17, no. 1, pp. 18–27, 2001.

[22] K. Toyama and E. Horvitz, "Bayesian modality fusion: Probabilistic integration of multiple vision algorithms for head tracking," in *Proc. of the Fourth Asian Conference on Computer Vision (ACCV)*, Taipei, 2000.

[23] H. Kruppa and B. Schiele, "Hierarchical combination of object models using mutual information," in *Proc. of British Machine Vision Conference*, Manchester, UK, Sept. 2001.

[24] J. L. Mundy and C.-F. Chang, "Fusion of intensity, texture, and color in video tracking based on mutual information," in *Proc. of the 33rd Applied Imagery Pattern Recognition Workshop*, vol. 1, Los Alamitos, CA, USA, Oct. 2004, pp. 10–15.

[25] Y. Wu and T. S. Huang, "Robust visual tracking by integrating multiple cues based on co-inference learning," *International Journal on Computer Vision*, vol. 58, no. 1, pp. 55–71, 2004.

[26] X. Zhong, J. Xue, and N. Zheng, "Graphical model based cue integration strategy for head tracking," in *Proc. of British Machine Vision Conference*, vol. 1, Edinburgh, UK, Sept. 2006, pp. 207–216.

[27] S. Khan and M. Shah, "Object based segmentation of video using color, motion and spatial information," in *Proc. of IEEE Conf. on Comp. Vis. and Pattern Recog.*, Kauai, HI, USA, Dec. 2001, pp. 746–751.

[28] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of International Conference on Computer Vision*, Corfu, Greece, Sept. 1999, pp. 1150–1157.

[29] J. Bigun and G. H. Granlund, "Optimal orientation detection of linear symmetry," in *Proc. of International Conference on Computer Vision*, London, UK, June 1987, pp. 433–438.

[30] J. Bigun, T. Bigun, and K. Nilsson, "Recognition by symmetry derivatives and the generalized structure tensor," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 12, pp. 1590–1605, 2004.

[31] E. Maggio, F. Smeraldi, and A. Cavallaro, "Combining colour and orientation for adaptive particle filter-based tracking," in *Proc. of British Machine Vision Conference*, Oxford, UK, Sept. 2005, pp. 659–668.

[32] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online non-linear/non-gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[33] J. Liu, R. Chen, and T. Logvinenko, "A theoretical framework for sequential importance sampling and resampling," Stanford University, Department of Statistics, Tech. Rep., Jan. 2000. [Online]. Available: www.people.fas.harvard.edu/~junliu/TechRept/00folder/smcb.ps

[34] R. A. Jacobs, "What determines visual cue reliability?" *Trends in Cognitive Sciences*, vol. 6, no. 8, pp. 345–350, 2002.

[35] K. Nickels and S. Hutchinson, "Estimating uncertainty in SSD-based feature tracking," *Image Vision Comput.*, vol. 20, no. 1, pp. 47–58, 2002.

[36] D. Doermann and D. Mihalcik, "Tools and techniques for video performance evaluation," in *Proc. of IEEE International Conf. on Pattern Recognition*, vol. 4, Barcelona, Spain, Sept. 2000, pp. 167–170.

Emilio Maggio received the MSc degree in telecommunication engineering from the University of Siena, Italy in 2003. Since 2004 he is a PhD. student at the Electronic Engineering department of the Queen Mary University of London, United Kingdom. In 2003 he visited the Signal Processing Institute at the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, working as a guest research student in the area of video coding and signal representation with redundant dictionaries. His research interests are object tracking, classification, Bayesian filtering, sparse image and video coding. He is currently working on a research project on ”Automatic object tracking and categorization”. He has also served as a reviewer for IEEE Transactions on Circuits and Systems for Video Technology and the Workshop on Image Analysis for Multimedia Interactive Services. He was a member of the team winner of CSIDC 2002, the IEEE Computer Society 3rd Annual International Design Competition, by means of the project ”Blue Sign Translator”, an instant translator from English to deaf sign language. Twice, in 2005 and 2007, he was awarded a best student paper prize at ICASSP, the IEEE International Conference on Acoustics, Speech, and Signal Processing.

Fabrizio Smeraldi received a Laurea in Physics from the University of Genoa, Italy in 1996, and a PhD from EPFL, Switzerland, in 2000. Until 2002 he held a faculty position at the University of Halmstad, Sweden. He then joined Queen Mary, University of London (UK), where he is a Lecturer in the Department of Computer Science. His research interests are in the area of pattern recognition and machine learning.

Andrea Cavallaro received the M.Sc. degree (laurea summa cum laude) from the University of Trieste, 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 at the Image Processing Laboratory, University of Trieste, Italy, working on compression algorithms for very low bit-rate video coding and on digital image sequence de-interlacing. In 1997, he served the Italian Army as lieutenant at the 33rd Electronic Warfare Battalion in Treviso, Italy. From 1998 to 2003 he was a Research Assistant at the Signal Processing Laboratory, EPFL. He was a Work-package Leader for the EU projects ACTS Modest and IST art.live and is Principal Investigator in a number of UK Research Council and industry-sponsored projects. Since 2003, he has been a Lecturer at Queen Mary, University of London (QMUL), London, U.K. His main research interests are image and video analysis, multimedia signal processing and interactive media computing. He acts as reviewer for several leading international conferences and journals, and he is author of more than 60 papers, including five book chapters.

Dr. Cavallaro was awarded a Research Fellowship with British Telecommunications (BT) in 2004/2005; the Drapers' Prize for the development of Learning and Teaching in 2004; an e-learning Fellowship in 2006; and the Royal Academy of Engineering teaching Prize in 2007. He is co-author of the papers ”Hybrid particle filter and mean shift tracker with adaptive transition model” and ”Particle PHD filtering for multi-target visual tracking”, winner of the student paper contest at the IEEE ICASSP in 2005 and 2007, respectively. Dr. Cavallaro has been a member of the organizing/technical committee of several conferences, including IEEE ICME, IEEE ICIP, SPIE VCIP, PETS, ACM Multimedia, IEEE AVSS, ECCV-VS; Guest Editor of the Special Issue on 'Multi-sensor object detection and tracking' (2007), Signal, Image and Video Processing Journal and co-Guest Editor of the Special Issue on 'Video Tracking in Complex Scenes for Surveillance Applications' (2008), Journal of Image and Video Processing. He is an elected member of the IEEE Signal Processing Society, Multimedia Signal Processing Technical Committee; General Chair of the IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS 2007); Chair of the 2007 BMVA symposium on Security and Surveillance; and Technical co-chair of the European Signal Processing Conference (EUSIPCO 2008).