Accurate appearance-based Bayesian tracking for maneuvering targets

Emilio Maggio, Andrea Cavallaro

Multimedia and Vision Group, Queen Mary University of London, Mile End Road, London E1 4NS (UK)

Abstract

We propose a tracking algorithm that combines the Mean Shift search in a Particle Filtering framework and a target representation that uses multiple semi-overlapping color histograms. The target representation introduces spatial information that accounts for rotation and anisotropic scaling without compromising the flexibility typical of color histograms. Moreover, the proposed tracker can generate a smaller number of samples than Particle Filter as it increases the particle efficiency by moving the samples toward close local maxima of the likelihood using Mean Shift. Experimental results show that the proposed representation improves the robustness to clutter and that, especially on highly maneuvering targets, the combined tracker outperforms Particle Filter and Mean Shift in terms of accuracy in estimating the target size and position while generating only 25% of the samples used by Particle Filter.

Key words: Object representation, color histogram, tracking, Mean Shift, Particle Filter

1 Introduction

Image-based tracking is an important component in many applications, such as video surveillance, medical image sequence analysis, augmented reality, smart rooms and object-based video compression. The goal of image-based tracking is to estimate the position and the shape of an object or a region over time. This

Email addresses: emilio.maggio@elec.qmul.ac.uk (Emilio Maggio), andrea.cavallaro@elec.qmul.ac.uk (Andrea Cavallaro).

[1] The authors acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC), under grant EP/D033772/1.

requires the definition of a target model and of a process that first generates candidate targets and then evaluates the similarity between the model and a candidate.

A simple and widely used *target model* is the template [1], which stores luminance or color values, and their location. Although template computation is simple and fast, the values stored in the template may become non-representative of the object appearance in presence of noise, partial occlusions, pose or scale changes. Solutions have been proposed to update the template over time [2–4] and to cope with occlusions [5] and pose changes [6]. However the complete pixel information may be unnecessary for the tracking task: a target representation should be descriptive enough to disambiguate the object from the background, while allowing a certain degree of flexibility to cope with changes of target scale, pose, scene illumination and partial occlusions. To this extent, color histograms have been used as target models for their invariance to scaling and rotation, robustness to partial occlusions, data reduction and efficient computation [7–10]. However, the descriptiveness of color histograms is limited by the lack of spatial information, which makes it difficult to discriminate targets with similar color properties. To overcome this problem, the information of the first two spatial moments associated to the location of the related color can be added to each bin of the histogram [11]. Alternatively, multiple histograms on different parts of the target can be used [8,12,13], although there is no widely accepted solution. The multi-part representation in [12] divides a target into two non-overlapping areas (top and bottom parts). This solution is effective for the specific application (i.e., tracking ice-hockey players), as it generally corresponds to the shirt and the trousers, but it is not necessarily effective on a generic target. An alternative to improve the distinctiveness of the target model is the use of multiple features. For example, gradient information can be used to complement color information [9,14,15]. However, computing several features for each candidate target may be computationally expensive for real-time applications.

After the definition of a target model, a *search method* is needed to select the candidate target locations to be evaluated against the model. To this extent, Particle Filters (PF) have been widely used in image-based tracking [16,8,17,2]. PF is a probabilistic method based on Monte Carlo sampling that can deal with multi-modal probability density functions (*pdf*s). PF-based trackers use the multiple hypotheses associated with the samples (i.e., the particles) to cope with occlusions and to recover from lost tracks. As the number of particles required to model the underlying *pdf* increases exponentially with the dimensionality of the state space, efficient proposal distributions for particle sampling are desirable. A popular choice is to draw the samples according to the target dynamic model, thus resorting to an algorithm known as CONDENSATION [18,16] (here referred to as PF-C). However, sampling in PF-C does not account for information from the most recent measurement. As a consequence, when the dynamic model is not accurate, the area of the state space around the target is not densely sampled. To account for the latest measurement many sampling strategies have been proposed [19–23]. Markov Chain Monte Carlo (MCMC) samplers have been used to sample the particles in high-dimensional state spaces (e.g., 3D body tracking) [20,24,25]. However, due to the relatively large number of steps necessary for MCMC to converge, no improvements in terms of efficiency are reported on low-dimensional state spaces [20]. Simulated Annealing is an alternative approach for particle sampling [26]: first particles are randomly spread over the state space, then a layered procedure re-draws the samples proportionally to their likelihood. When the relationship between state and measurement can be linearized, an alternative is to sample from the Gaussian estimate computed by an Extended Kalman Filter associated to each particle [19]. Similarly, EKF can be substituted with an unscented transform that does not require linearization [19,27]. Both methods assume that the modes of the *pdf* are well represented by their first and second order moments. Furthermore, while PF-C requires explicit definition of the likelihood only, the last two methods also require the explicit formulation of the measurement equation.

A different approach to particle sampling is to drive the particles according to point estimates of the gradient of either the posterior or the likelihood [21,20,24,25,28,29]. When the appearance model is a template, optical flow can be used to drive the particles towards peaks of the likelihood. However, as motion blur can affect the accuracy of optical flow, the particle shifting procedure is enabled only when the momentum of the object is small [21]. A more principled solution, known as Kernel Particle Filter, uses kernel density estimation to produce from the particle set a continuous approximation of the posterior *pdf*. Then, sample-based Mean Shift (MS), a kernel-based iterative procedure, is used to approximate the gradient of the *pdf* and to climb its modes [22,23]. However, as the accuracy of density estimate and of its gradient depends on the sampling rate, a reduction of the number of samples may affect the quality of the final approximation. An alternative to sample-based MS is color-based MS [7], a very popular tracking algorithm that uses color histograms. Color-based MS performs a gradient descent of the model-candidate distance using the kernel-weighted color density (and not the sample density) estimate. Unlike PF, which requires the costly computation of one candidate model (e.g., a color histogram) per particle, color-based MS has a low computational cost. In fact the gradient estimate requires the computation of one histogram per iteration only. However, while sample-based approaches such as PF are flexible in terms of search region, the search of color-based MS is limited by the kernel size. For this reason, color-based MS fails to track small and fast moving targets as well as to recover the position of a target after a total occlusion. Given the complementarity of the two algorithms, combinations of MS and PF have been studied [28,29]. However, the convergence of the MS procedure with the used flat kernel is not demonstrated [28]. Moreover,

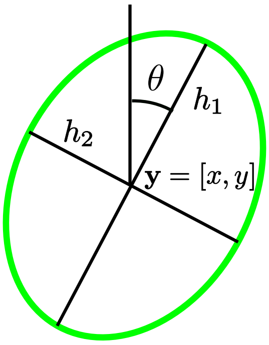


Fig. 1. Parameters defining the ellipse bounding the target area.

as the particles are re-displaced by MS, it is not clear how the new sampling distribution can be approximated to correctly compute their weights [28,29].

In this paper we propose an algorithm that improves the robustness of the target representation and increases tracking flexibility and effectiveness. The target representation uses semi-overlapping color histograms that improve the sensitivity to rotations and anisotropic scale changes, while maintaining the robustness and flexibility typical of single color histograms. Related to this partition, an extension of MS is proposed, which is used to find local minima of the model-candidate distance in the state space. Particles generated by PF are shifted toward these minima, which represent positions with high probability of locating a target, thus increasing the efficiency of the particles. The increased efficiency is obtained on low-dimensional state spaces (3-D to 5-D), where other gradient-based sampling methods are ineffective [20].

The paper is organized as follows. Section 2 discusses the use of spatial information in target representation and introduces the color histogram representation. The proposed tracker is presented in Sec. 3. Section 4 describes the evaluation procedure used in Sec. 5 to assess the experimental results. Finally, in Sec. 6 we draw conclusions and discuss future research directions.

2 Target Representation

Let us approximate the target shape with an ellipse and represent the target state with $\mathbf{x} = [\mathbf{y}, \mathbf{s}]$, where $\mathbf{y} = [x, y]$ is the center of the ellipse and $\mathbf{s} = [e, \theta, h_1]$. The variable e is the ellipse eccentricity, θ is its clockwise rotation and h_1 is the length of the semi-axis used as reference for the rotation. In the following we will also use h_2 as the length of the second semi-axis (Fig. 1).

Let the target representation of the pixels inside the ellipse be their weighted color distribution approximated by a normalized color histogram. Given an image $\mathbf{z}$ the normalized color histogram $\mathbf{r}(\mathbf{x}, \mathbf{z}) = \{r_u(\mathbf{x}, \mathbf{z})\}_{u=1,\ldots,U}$ with U

bins of a target candidate $\mathbf{x}$ can be calculated by selecting for each pixel in the ellipse the bin index u corresponding to its color and then cumulating on the bin the values obtained with a weighting kernel $k(.)$. The kernel $k(.)$ usually gives higher weight to pixels near the center of the ellipse as they are less likely to be occluded by other objects [7]. Given the coordinates of the $n(\mathbf{x})$ pixels inside the ellipse $\{\mathbf{w}_i\}_{i=1,\ldots,n(\mathbf{x})}$, the Dirac's delta function $\delta(.)$, a function $b(\mathbf{w}_i, \mathbf{z})$ that associates a pixel of the image $\mathbf{z}$ with position $\mathbf{w}_i$ to the histogram bin, and

$$A(\mathbf{s}) = \begin{bmatrix} \frac{\cos\theta}{h_2} & -\frac{\sin\theta}{h_2} \\ \frac{\sin\theta}{h_1} & \frac{\cos\theta}{h_1} \end{bmatrix},$$

the matrix used to scale and rotate the kernel, the computation of the bin value $r_u(\mathbf{x}, \mathbf{z})$ can be formalized as

$$r_u(\mathbf{x}, \mathbf{z}) = C(\mathbf{x}) \sum_{i=1}^{n(\mathbf{x})} k\left(\|A(\mathbf{s})(\mathbf{y} - \mathbf{w}_i)\|^2\right) \delta\left[b(\mathbf{w}_i, \mathbf{z}) - u\right] \quad u = 1, \ldots, U; \quad (1)$$

where and $C(\mathbf{x})$ is a normalization function defined as

$$C(\mathbf{x}) = \frac{1}{\sum_{i=1}^{n(\mathbf{x})} k\left(\|A(\mathbf{s})(\mathbf{y} - \mathbf{w}_i)\|^2\right)}. \quad (2)$$

Then, we define the target model as the color distribution of the object at track initialization, i.e., $\mathbf{o} = \mathbf{r}(\mathbf{x}_I, \mathbf{z}_I)$, where $\mathbf{x}_I$ and $\mathbf{z}_I$ are the state and the image frame at initialization.

The matching quality of a candidate is defined by the candidate-model distance, d, between the normalized histograms $\mathbf{r}(\mathbf{x}, \mathbf{z})$ and $\mathbf{o}$:

$$d\left[\mathbf{r}(\mathbf{x}, \mathbf{z}), \mathbf{o}\right] = \sqrt{1 - \rho\left[\mathbf{r}(\mathbf{x}, \mathbf{z}), \mathbf{o}\right]}, \quad (3)$$

where ρ is the Bhattacharyya coefficient [30]

$$\rho\left[\mathbf{r}(\mathbf{x}, \mathbf{z}), \mathbf{o}\right] = \sum_{u=1}^{U} \sqrt{r_u(\mathbf{x}, \mathbf{z}) \cdot o_u}. \quad (4)$$

Despite their success in image-based tracking, as color histograms do not encode spatial information, errors are likely to happen, as shown in the example of Fig. 2. The tracker is attracted to false targets with similar color properties, such as the shadow of a car (similar to the trousers) and the white car (similar to the shirt). Some knowledge of the color spatial distribution is needed to overcome this limitation. For this reason we propose a simple yet effective and general solution for target representation based on multiple histograms calculated over semi-overlapping regions. This representation incorporates global and local target information in a single model. The first histogram is calcu-

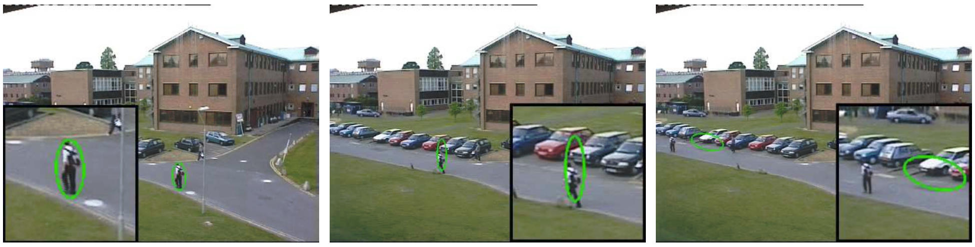


Fig. 2. Example of failure due to the target representation using a single color histogram.

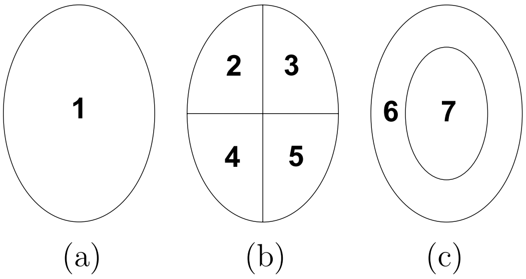


Fig. 3. Semi-overlapping histogram-based target representation. (a) Single histogram of the whole target, (b) rotation-sensitive partition, (c) size-sensitive partition.

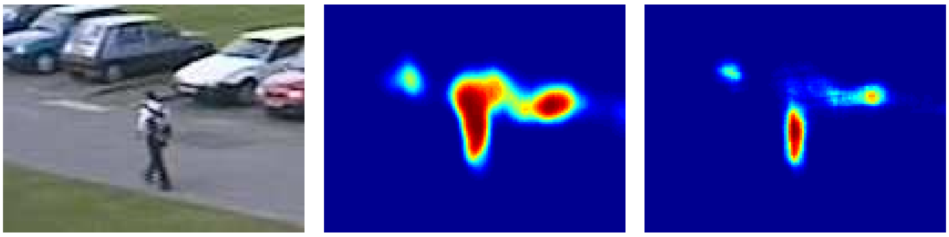
lated over the whole target (Fig. 3(a)). To account for rotations, four parts are then obtained from the partition created by the two axes (Fig. 3(b)). Finally, to account for scale changes, the inner and outer areas of a concentric ellipse with same eccentricity, but half axis size than the whole ellipse, are considered (Fig. 3(c)). The proposed target model is referred to as Multi-Part model (MP).

Eq. (4) can be extended to multiple histograms as

$$\rho_{MP}\left[\mathbf{r}(\mathbf{x},\mathbf{z}),\mathbf{o}\right] = \frac{\sum_{j=1}^{N}\rho\left[\mathbf{r}_j(\mathbf{x},\mathbf{z}),\mathbf{o}_j\right]}{N}, \tag{5}$$

where N is the number of parts (in our case, $N = 7$), $\mathbf{r}_j$ and $\mathbf{o}_j$ are the model and candidate histograms calculated on the j^{th} part. The model-candidate distance is computed as in Eq. (3), using Eq. (5).

Fig. 4 visualizes the values of the model-candidate coefficients of Eq. (4) and Eq. (5) for the problem described in Fig. 2. It is possible to notice that the area with high model-candidate matching (red area) is narrower for MP (Fig. 4 (c)), thus reducing the probability of attraction to the false target.



(a) (b) (c)

Fig. 4. Comparison of the model-candidate Bhattacharyya coefficients ρ using the single color histogram and the coefficients ρ_{MP} using the proposed multi-part (MP) representation. (a) Sample frame; (b) ρ; (c) ρ_{MP}. Red indicates higher model-candidate similarity.

3 Target tracking

3.1 Target state as a particle

After the definition of a target model (Sec. 2), a method is needed to search for candidate objects based on the previous states and on the current image. Using PF, the tracking problem can be addressed using the dynamical and measurement equations [16]

$$\mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_{t-1}, \mathbf{v}_t), \tag{6}$$

$$\mathbf{z}_t = \mathbf{h}_t(\mathbf{x}_t, \mathbf{n}_t), \tag{7}$$

where $\mathbf{x}_t$ is the state at time t, $\mathbf{f}_t$ and $\mathbf{h}_t$ are non-linear time-varying functions, $\{\mathbf{v}_t\}_{t=1,\ldots}$, $\{\mathbf{n}_t\}_{t=1,\ldots}$ are assumed to be independent and identically distributed stochastic processes and the measurement $\mathbf{z}_t$ is the current image frame captured by the camera sensor. The *pdf* $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ is estimated recursively in two steps, namely prediction and update. The *prediction step* uses Eq. (6) to obtain the predicted *pdf* as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})d\mathbf{x}_{t-1}, \tag{8}$$

with $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})$ known from the previous iteration and $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ determined by Eq. (6). The *update step* uses the Bayes' rule once the measurement $\mathbf{z}_t$ is available:

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{\int p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})d\mathbf{x}_t}. \tag{9}$$

The densities $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ are approximated with a sum of N_s Dirac functions centered in $\{\mathbf{x}_t^i\}_{i=1,\ldots,N_s}$ as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \approx \sum_{i=1}^{N_s} \omega_t^i \delta\left(\mathbf{x}_t - \mathbf{x}_t^i\right), \tag{10}$$

where ω_t^i, the weights associated to the particles, are

$$\omega_t^i \propto \frac{p(\mathbf{x}_t^i|\mathbf{z}_{1:t})}{q(\mathbf{x}_t^i|\mathbf{z}_{1:t})}. \tag{11}$$

$q(.)$ is the importance density function defined as the density that generated the current set of particles.

Assume that $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})$ is approximated by the set of particles $\left\{\mathbf{x}_{t-1}^j\right\}_{j=1,\ldots,N_s}$ as in Eq. (10). Substituting this approximation in Eq. (8) and then applying Eq. (9) we obtain

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \propto p(\mathbf{z}_t|\mathbf{x}_t) \sum_{j=1}^{N_s} \omega_{t-1}^j p(\mathbf{x}_t|\mathbf{x}_{t-1}^j). \tag{12}$$

The mixture of the transition probabilities weighted by the weights at time $t-1$ approximates the predicted *pdf* $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$. Given an importance sampling function $q(.)$, from Eq. (10) and Eq. (12) the updated set of weights $\{\omega_t^j\}_{i=1,\ldots,N_s}$ at time t is determined as

$$\omega_t^i \propto \frac{p(\mathbf{z}_t|\mathbf{x}_t^i) \sum_{j=1}^{N_s} \omega_{t-1}^j p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^j)}{q(\mathbf{x}_t^i|\mathbf{z}_{1:t})}. \tag{13}$$

3.2 Particle propagation

The particles are first drawn from the predicted *pdf* (i.e, $q(\mathbf{x}_t^i|\mathbf{z}_{1:t}) = p(\mathbf{x}_t^i|\mathbf{z}_{1:t-1})$) using a zero-order state transition model

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{v}_t, \tag{14}$$

where $\mathbf{v}_t$ is a multivariate Gaussian random variable with 0 mean vector and constant standard deviations. This is equivalent to performing a standard resampling procedure at each iteration and then propagating according to the transition model. A new predicted set of particles $\{\tilde{\mathbf{x}}_t^i\}_{i=1\ldots,N_s}$ is now available.

The choice of a relatively uninformative motion model (i.e., a zero–order autoregressive model) is motivated by the fact that we aim to track also highly maneuvering targets. Although a more complex motion model could improve tracking performance with targets having a predictable behavior, the same model would not be appropriate for highly maneuvering (unpredictable) targets. Unfortunately, with CONDENSATION the use of an uninformative motion model together with sampling from the predicted *pdf* $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ results in a largely suboptimal filter in terms of variance of the Bayes state estimator for a given number of particles [16]. In fact, if the motion model is uninformative, the posterior *pdf* can significantly differ from the prior (the *pdf* used by CONDENSATION to sample from). A sampling criterion that uses the current observation to distribute particles around regions of high likelihood is expected to increase the filter efficiency. To this end we propose a method based on the MS search, as described below.

3.3 Mean Shift

After propagation, each particle in $\{\tilde{\mathbf{x}}_t^i\}_{i=1\ldots,N_s}$ is independently re-located in the position state subspace using color-based MS [7]. This process iteratively minimizes the distance in Eq. (3) using gradient information. The algorithm is initialized at $\mathbf{x}_a = \tilde{\mathbf{x}}_t^i$, the particle position, and converges to the nearest local minimum by shifting at each iteration the particle centroid from $\mathbf{y}_a$ to $\mathbf{y}_b$. Given $g(z) = -k'(z)$ (i.e., $-g(.)$ is the first order derivative of the kernel $k(.)$), the weights

$$w_i = \sum_{u=1}^{U} \sqrt{\frac{o_u}{r_u(\mathbf{x}_a, \mathbf{z}_t)}} \delta\left[b(\mathbf{w}_i, \mathbf{z}) - u\right], \tag{15}$$

and

$$B(\mathbf{s}) = \begin{bmatrix} \left(\frac{h_1}{h_2}\cos\theta\right)^2 + (\sin\theta)^2 & \sin\theta\cos\theta\left(1 - \frac{h_1^2}{h_2^2}\right) \\ \sin\theta\cos\theta\left(1 - \frac{h_1^2}{h_2^2}\right) & \left(\frac{h_1}{h_2}\sin\theta\right)^2 + (\cos\theta)^2 \end{bmatrix}.$$

a correction matrix used to account for kernel rotation and anisotropic scaling, the new location $\mathbf{y}_b$ is defined as

$$\mathbf{y}_b = B(\mathbf{s}_a)\left(\frac{\sum_{i=1}^{n(\mathbf{x}_a)} \mathbf{w}_i w_i g\left(\|A(\mathbf{s}_a)(\mathbf{y}_a - \mathbf{w}_i)\|^2\right)}{\sum_{i=1}^{n(\mathbf{x}_a)} w_i g\left(\|A(\mathbf{s}_a)(\mathbf{y}_a - \mathbf{w}_i)\|^2\right)} - \mathbf{y}_a\right) + \mathbf{y}_a. \tag{16}$$

Eq. (16) extends the classical MS formulation [7] to anisotropic and rotated kernels. When $h_1 = h_2$ and $\theta = 0$ it was proved that $\mathbf{y}_b - \mathbf{y}_a$ is in the gradient direction [7]. For generic h_1, h_2 and θ demonstration is provided in the Appendix. The iterative process stops when $\|\mathbf{y}_b - \mathbf{y}_a\| < \epsilon$. Usually $\epsilon = 1$ pixel [7]. If the condition is not met $\mathbf{y}_a$ takes the value in $\mathbf{y}_b$ and another MS step is performed.

When multi-part histograms are used (Sec. 2), the coefficient to maximize is the ρ_{MP} defined in Eq. (5). We can derive a MS iterative procedure equivalent to Eq. (16) for multiple histograms using

$$\mathbf{y}_b = B(\mathbf{s}_a) \left(\frac{\sum_{j=1}^{N} C_j(\mathbf{x}_a) \sum_{i=1}^{n_j(\mathbf{x}_a)} \mathbf{w}_{j,i} w_{j,i} g \left(\| A(\mathbf{s}_a) \left(\mathbf{y}_a - \mathbf{w}_{j,i} \right) \|^2 \right)}{\sum_{j=1}^{N} C_j(\mathbf{x}_a) \sum_{i=1}^{n_j(\mathbf{x}_a)} w_{j,i} g \left(\| A(\mathbf{s}_a) \left(\mathbf{y}_a - \mathbf{w}_{j,i} \right) \|^2 \right)} - \mathbf{y}_a \right) + \mathbf{y}_a, \tag{17}$$

where $w_{j,i}$ is the coefficient defined in Eq. (15) for the j^{th} histogram and the contribution of each part is weighed by the normalization factor $C_j(\mathbf{x}_a)$. It is possible to demonstrate that the multi-part MS steps, similarly to the single histogram case [7], are in the direction of the gradient of ρ_{MP}; the validity of this result is independent from the partition (for the demonstration, see the Appendix).

Although it is possible to modify Eq. (17) to use different kernels on different parts, to reduce computational cost we use a single kernel with Epanechnikov profile [31], that is

$$k(z) = \begin{cases} 1 - z & \text{if } z < 1 \\ 0 & \text{otherwise} \end{cases} . \tag{18}$$

The weight of each pixel is proportional to the distance from the ellipse centroid and is independent on the shape of the partition. As the derivative $k'(.)$ is constant Eq. (17) reduces to

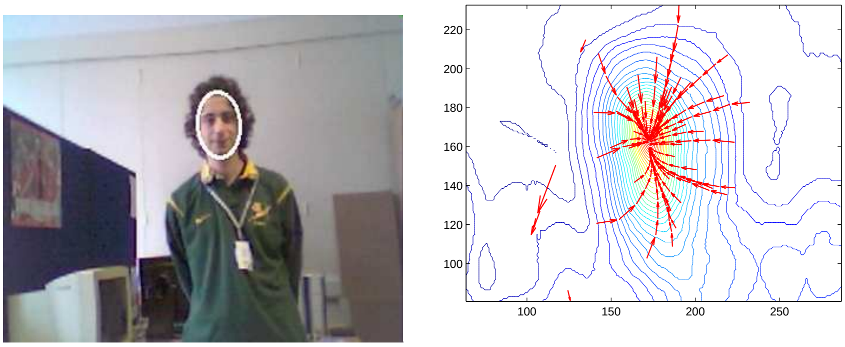
$$\mathbf{y}_b = B(\mathbf{s}_a) \left(\frac{\sum_{j=1}^{N} C_j(\mathbf{x}_a) \sum_{i=1}^{n_j(\mathbf{x}_a)} \mathbf{w}_{j,i} w_{j,i}}{\sum_{j=1}^{N} C_j(\mathbf{x}_a) \sum_{i=1}^{n_j(\mathbf{x}_a)} w_{j,i}} - \mathbf{y}_a \right) + \mathbf{y}_a. \tag{19}$$

Fig. 5 shows an example of convergence of the multi-part coefficient of Eq. (5) to a local maximum. In the example the target is a face and 40 different initializations are selected. The corresponding paths of the MS procedure are shown with red arrows. The MS procedures initialized on the mode of the Bhattacharyya coefficient generated by the target converge to the actual target location.

To summarize, let MS be defined as $\mathcal{M} : R^D \rightarrow R^D$, where D is the state space dimensionality. The final set of particles $\{\mathbf{x}_t^i\}_{i=1...,N_s}$ is obtained by applying

$$\mathbf{x}_t^i = \mathcal{M}(\tilde{\mathbf{x}}_t^i), \quad i = 1, \ldots, N_s. \tag{20}$$

$\mathcal{M}$ takes each particle $\tilde{\mathbf{x}}_t^i$ as input and modifies the state position, $\mathbf{y}$, guiding each particle over the position sub-space independently from all others.



(a) (b)

Fig. 5. Multi-part Mean Shift iterations using semi-overlapping color histograms. (a) Target; (b) Mean Shift vectors superimposed on the isolevel curves representing the value of the Bhattacharyya coefficient ρ_{MP}.

3.4 Weighting and state estimation

To compute the particle weights, we derive the likelihood $p(\mathbf{z}_t|\mathbf{x})$ from the distances of Eq. (3) as

$$p(\mathbf{z}_t|\mathbf{x}) = \exp\left\{-\frac{d\left[\mathbf{r}(\mathbf{x},\mathbf{z}_t),\mathbf{o}\right]^2}{\sigma^2}\right\}, \tag{21}$$

The value of σ depends on the histogram dimensionality. The higher the dimensionality, the larger the average distance of Eq. (3), hence the higher the value for σ used to obtain a smoother likelihood.

In CONDENSATION (PF-C) [16], $q(\mathbf{x}_t|\mathbf{z}_{1:t}) = p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$, is the predicted prior (Sec. 3.3), thus the weighting reduces to

$$\omega_t^i \propto p(\mathbf{z}_t|\mathbf{x}_t^i), \tag{22}$$

i.e., the weights are proportional to the likelihood. As the particles are shifted by the MS procedure, the importance sampling function $q(.)$ is no more the predicted prior. Weighting according to Eq. (22) would introduce a bias in the posterior approximation. To prevent this we approximate, as in Kernel Particle Filter [22], $q(\mathbf{x}_t|\mathbf{z}_{1:t}) \approx \hat{q}(\mathbf{x}_t)$ using a Gaussian kernel density estimation as

$$\hat{q}(\mathbf{x}_t) = \frac{1}{N_s}\sum_{i=1}^{N_s}\hat{q}_i(\mathbf{x}_t), \tag{23}$$

where $\hat{q}_i(\mathbf{x}_t)$ is the Gaussian kernel with smoothing bandwidth β defined as

$$\hat{q}_i(\mathbf{x}_t) = Z(\beta,\hat{\Sigma}_t)\cdot\exp\left(-\frac{1}{2\beta^2}\left(\mathbf{x}_t-\mathbf{x}_t^i\right)^T\hat{\Sigma}_t^{-1}\left(\mathbf{x}_t-\mathbf{x}_t^i\right)\right). \tag{24}$$

T denotes the transpose and

$$Z(h, \hat{\Sigma}_t) = \frac{1}{\left(\beta\sqrt{2\pi}\right)^D \sqrt{|\hat{\Sigma}_t|}}. \tag{25}$$

D is the dimensionality of the state space, and $\hat{\Sigma}_t$ is the covariance matrix of the particles state parameters computed as

$$\hat{\Sigma}_t = \frac{1}{N_s}\sum_{i=1}^{N_s}(\mathbf{x}_t^i - \bar{\mathbf{x}})(\mathbf{x}_t^i - \bar{\mathbf{x}})^T,$$

where $\bar{\mathbf{x}}$ is the particle mean. In order to select the kernel bandwidth β, we use a result from Silverman [32]. The value of β that gives the optimal rate of convergence in probability to zero of the integrated squared error $\int \left(\hat{q}(\mathbf{x}_t) - q(\mathbf{x}_t)\right) d\mathbf{x}_t$ is

$$\beta = c \cdot N_s^{\frac{-1}{(D+4)}}, \tag{26}$$

where the constant c is $c = \left(\frac{4}{D+2}\right)^{1/(D+4)}$.

Finally, the best state at time t is estimated from the discrete approximation of Eq. (10). The most common solution is to use the Bayes Least Squares estimate defined as

$$\mathbb{E}[\mathbf{x}_t|\mathbf{z}_{1:t}] \approx \frac{1}{N_s}\sum_{i=1}^{N_s}\omega_t^i\mathbf{x}_t^i. \tag{27}$$

3.5 Discussion

The hybrid algorithm (here referred to as HY) presents several advantages compared to MS and PF-C. In MS, the target search is limited to the image area spanned by the kernel (Eq. (16)). For this reason, if the shift of the target center is larger than the kernel size, the track is likely to be lost. Also, MS minimizes ρ with respect to the centroid estimate $\mathbf{y}$ but does not estimate the other parameters in $\mathbf{s}$. HY overcomes both these problems thanks to the multiple MS initializations generated by the particles. HY extends the volume of the state space under analysis to the space spanned by the particles as defined by the importance sampling function $q(.)$. Also, the extra three state parameters in $\mathbf{s}$ describing target rotation and anisotropic scaling are estimated by HY as in PF-C. Furthermore, HY inherits from PF-C the possibility to treat multi-modal *pdf*s and to recover from short-term occlusions. However, unlike PF-C, HY operates on particles that are concentrated near local maxima of the likelihood (Fig. 5). In Sec. 5 we will show how this property makes the hybrid algorithm more efficient than PF-C.

To conclude, we highlight the difference between the proposed approach and

Kernel Particle Filter (KPF) [22], as both algorithms move the particles using MS. KPF estimates the MS vector in an arbitrary state x from a set of kernels centered on the position of the particles. As the contribution of each particle depends on its distance from x, then the estimate depends on the particle configuration and can be critical when the number of particles is small. Unlike KPF, the proposed approach uses the color-based version of MS [7] where the kernels are centered on the single pixel positions (Eq. (16)). Although more specific to the appearance-based tracking problem, the proposed algorithm shifts each particle independently from the configuration of the other samples and consequently the shift is not influenced by the sample density.

4 Objective evaluation

The quality of the tracking results is evaluated by a combination of visualization and objective measures. Objective evaluation measures have attracted a considerable interest [33–39] and are used for the final assessment of an algorithm as well as for the development process to compare different parameter sets of an algorithm on large datasets.

The properties to be evaluated in tracking results are the error of the estimated target shape, the error of the estimated target position, and the percentage of lost tracks. We associate to each property a performance measure, namely *dice*, the *normalized centroid error*, and the *lost track ratio*.

Dice, $\mathcal{D}(t)$, measures in each frame t the match between set of pixels $A_e(t)$ and $A_g(t)$, defined by the estimated and ground-truth ellipses, as

$$\mathcal{D}(t) = 1 - \frac{2|A_e(t) \cap A_g(t)|}{|A_e(t)| + |A_g(t)|}, \tag{28}$$

where $|.|$ denotes the cardinality of a set. $\mathcal{D}(t)$ rewards candidates with a high percentage of true positive pixels, and with few false positives and false negatives.

The *normalized centroid error*, $\eta(t)$, evaluates the precision of the algorithm in estimating the centroid of a target, normalized by the target size. Let $l_{1,g}$, $l_{2,g}$, and θ_g be the length of the two semi-axes and the rotation of the ground-truth target area. The normalized error in x and y is then defined as

$$\epsilon_x = \frac{\cos\theta_g(x_e - x_g) - \sin\theta_g(y_e - y_g)}{l_{1,g}} \tag{29}$$

and

$$\epsilon_y = \frac{\sin\theta_g(x_e - x_g) + \cos\theta_g(y_e - y_g)}{l_{2,g}}, \tag{30}$$

where (x_e, y_e) and (x_g, y_g) are the centroid coordinates of the estimated and ground truth ellipses respectively. The normalized Euclidean error at frame t is

$$\eta(t) = \sqrt{\epsilon_x(t)^2 + \epsilon_y(t)^2}. \tag{31}$$

If the estimated centroid is outside the ground-truth area, then $\eta(t) > 1$.

The *lost track ratio*, λ, is the ratio between the number of frames where the tracker is unsuccessful and the target life span. Using Eq. (28), a lost track at t is declared when $\mathcal{D}(t) > 0.85$.

The above measures are used in the final *performance vector* $(\lambda, \bar{\mathcal{D}}, \bar{\eta})$, which is composed of λ; the average value of $\mathcal{D}(t)$ over the frames where the track is not lost, $\bar{\mathcal{D}}$; and the average value of $\eta(t)$ over the frames where the track is not lost, $\bar{\eta}$. Care is necessary while analyzing the numerical results as the values of $\bar{\mathcal{D}}$ and $\bar{\eta}$ are dependent on λ. A lower λ means that $\bar{\mathcal{D}}$ and $\bar{\eta}$ are evaluated on a larger number of frames; the added frames are usually in more challenging portions of a sequence. Therefore, the value of λ has to be analyzed first, and then, if the values obtained by two trackers are similar, we analyze the other two performance measures.

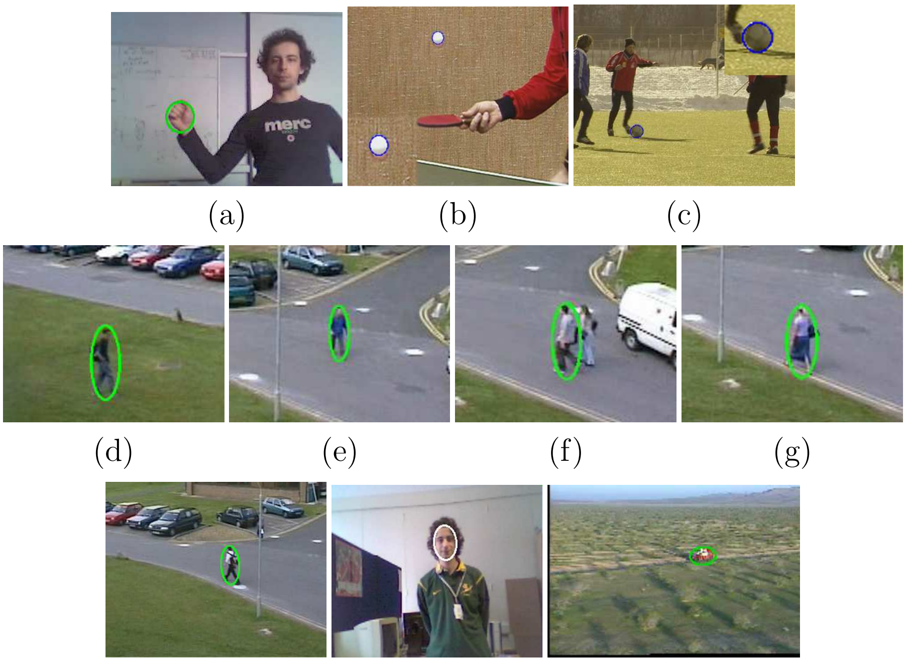
To evaluate stochastic algorithms like PF, we calculate the average $(\lambda_R, \bar{\mathcal{D}}_R, \bar{\eta}_R)$ and standard deviation $(\sigma(\lambda_R), \sigma(\bar{\mathcal{D}}_R), \sigma(\bar{\eta}_R))$ over R runs for each target in each sequence of the dataset. A good tracker should have consistent results on all the runs (i.e., low standard deviation). Each measure is also averaged over the K targets in a sequence

$$E_R = \frac{1}{R}\sum_{r=1}^{R} \frac{\sum_{j=1}^{K} F_j \cdot e_j}{\sum_{j=1}^{K} F_j}, \tag{32}$$

where e_j is one of the components of $(\lambda, \bar{\mathcal{D}}, \bar{\eta})$ for the target j that is weighted proportionally to the number of visible frames F_j.

5 Results

In this section we compare the results of the proposed Hybrid tracker (HY) with MS and PF-C. We test these three algorithms with the classic single color histogram and with the multi-part target representation (MS-MP, PF-C-MP, HY-MP).



(a) (b) (c)

(d) (e) (f) (g)

(h) (i) (j)

Fig. 6. The targets of the dataset used for the evaluation. A hand from sequence *S1* (a), a table tennis ball from sequence *S2* (b); a football from sequence *S3* (c); 5 pedestrians from sequence *S4* (d)-(h); a face from sequence *S5* (i); a vehicle from sequence *S6* (j).

5.1 Dataset

We evaluated the trackers on a dataset containing 10 heterogeneous targets[2] extracted from 6 sequences (Fig. 6 and Table 1). The total number of target frames used in the evaluation is 5098. Three out of the six sequences (i.e., *S2*, *S3* and *S6*) are shot with non-static cameras.

Although the sequences present more than the selected moving objects, for the purpose of objective evaluation we manually annotated targets presenting different motion behaviors and different levels of distinctiveness with respect to the background. The dataset includes three sequences (*S1*, *S2* and *S3*) with fast moving targets. In *S1*, shot with a low-quality web camera, a hand performs abrupt and unpredictable movements. Also, the tracker can be misled by skin areas other than the hand, like the face, with similar color properties. In *S2* (the MPEG-4 test sequence *Table tennis*), the target is a table tennis ball with fast speed changes. In *S3* (the MPEG-4 test sequence *Soccer*), the aim is to track a football. An abrupt acceleration is caused by a player kicking the

[2] The ground-truth annotation for all the targets of the dataset, the test sequences and sample results are available at `http://www.elec.qmul.ac.uk/staffinfo/andrea/HY-MP.html`

Table 1
Description of the tracking dataset. Frame sizes are in pixels. The target size is the number of pixels inside the ground-truth ellipse.

Seq.	**Frame size**	**Frame rate (Hz)**	**Target size**		**Tracking challenges**	**Static camera**
			Min	Max		
S1	320 × 240	12.5	1004	2725	Highly maneuvering, clutter	Yes
S2	352 × 288	24	131	310	Highly maneuvering	No
S3	352 × 288	30	314	452	Highly maneuvering, occlusion	No
S4	768 × 576	25	282	2543	Clutter, occlusions	Yes
S5	384 × 258	25	942	18362	Clutter, occlusion	Yes
S6	352 × 240	25	125	3306	Aerial camera	No

ball, which is then occluded in two instances by the legs of the players for 3 and 21 frames, respectively. Five pedestrians are extracted from *S4* (PETS2001, dataset 1, sequence *camera1*). The targets share similar color properties with some parked cars in the background. Also, the pedestrians are briefly occluded by a lamp post. Finally, the images are affected by high level of sensor and compression noise. Sequence *S5*, shot with a low quality camera, presents a face tracking scenario where a desk partially occludes the target for 22 frames and clutter is generated by a bookshelf in the background. Finally, one off-road vehicle is extracted from the aerial sequence *S6*, *Redteam* [40].

5.2 Testing conditions

To enable a fair comparison, each algorithm under analysis is initialized manually using the first target position defined by the ground-truth annotation. The parameter setting is described in the following. Color histograms are calculated in the RGB space quantized with 8x8x8 bins and MS runs 5 times with different kernel sizes up to +/-10% than the previous frame. We test the algorithms with two state models with 3 dimensions (3D) and 5 dimensions (5D), respectively. The 3D state model is composed of target position, (x, y), and target size h_1. The 5D state model is composed of eccentricity, e, and rotation, θ, in addition to position and size. PF-C and HY use the 3D and the 5D state models; while MS, uses the 3D state model only, as in [7]. The Gaussian random variable $\mathbf{v}_t$ has standard deviations $\sigma_x = \sigma_y = 7$ pixels, $\sigma_{h_1} = 0.07\%$, $\sigma_e = 0.03$, and $\sigma_\theta = 5.0^o$. Note that the scale change is a percent and the angle is given in degrees. We chose the values of σ_x and σ_y as a compromise between slow and fast targets. PF-C uses 150 samples in the 3D case and 250 in the 5D case. HY uses 25% of the samples used by PF-C. To limit the computational cost of HY and to avoid particle degeneration (due to MS all the particles converge to similar states), the number of MS iterations

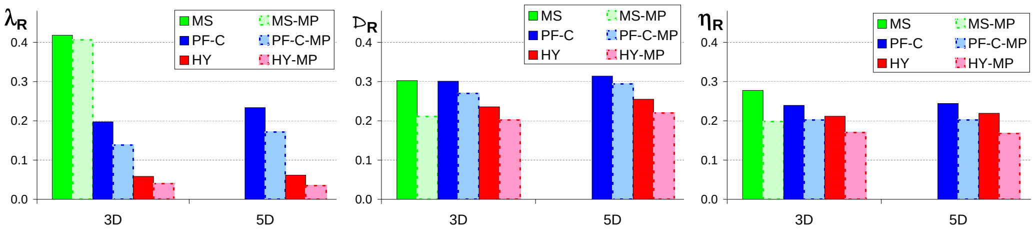


Fig. 7. Comparison of tracking results for the proposed algorithm (HY) and for the proposed target representation (MP suffix). The bar plots show the average performance scores over the targets in the evaluation dataset. HY lowers the errors with respect to MS and PF-C for all the performance scores. Also, MP improves the tracking result for all the three algorithms.

in HY is limited to 3 for each particle.

5.3 Evaluation

Figure 7 summarizes the results for the three algorithms under analysis (i.e., MS, PF-C and HY) with and without the proposed multi-part target representation (MP), by displaying the average scores over the whole dataset. The complete results for each sequence are available in Table 2. HY greatly reduces the lost tracks (lower λ_R) with respect to MS and PF-C. Also, the state estimates produced by HY are in average more accurate than those of PF-C and MS (Fig. 7). More in detail, from Table 2 we note that HY outperforms MS all over the dataset, except for *S6* ($\bar{\mathcal{D}}_R$ is 0.26 and 0.28 for MS and HY, respectively), thus confirming the stability of MS on targets with a limited motion. Compared with PF-C, a clear advantage of HY is when the state transition model does not predict correctly the behavior of the target. In PF-C particles are denser around the previous state position. The faster the target, the smaller the density of the particles around it. HY eliminates this problem using the MS procedure, leading to a largely improved performance for *S1*, *S2*, and *S3*. As discussed in Sec. 4, the low value of $\bar{\eta}_R$ retuned by PF-C in *S3* has to be disregarded due to the large performance gap indicated by λ_R. HY shows similar performance to PF-C in tracking slow targets like these in *S4*, *S5*, and *S6*. The results of Table 2, and the observation that HY uses 75% less particles than PF demonstrates our claim on the improved sampling efficiency of HY compared to PF-C.

By comparing the results of the 5D and 3D state models (Fig. 7) it is interesting to observe that adding extra degrees of freedom does not improve the results: HY achieves similar performance on the two models, while PF-C with the 5D state is worse than with the 3D state. In fact, although extra state parameters should add flexibility to the ellipse fitting process, this flexibility requires a larger number of particles and is counterproductive when the appearance model is not sufficiently discriminative, for example when the target appearance is similar to the background.

Sample results from the test sequences *S2* and *S3* are shown in Fig. 8. The targets are moving in unexpected directions with shifts larger than the kernel size. Moreover, the targets are affected by motion blur that decreases the effectiveness of the MS vector. HY is more stable in maintaining the track of the balls (Fig. 8 (a)-(c)) and reduces by 75% and 83% the value of λ_R (Table 2) for the two ball targets, respectively. In *S2*, PF-C recovers the target after losing it, but then it fails again. In *S3*, MS and PF are not able to track the ball, whereas HY is fast in reacting to the abrupt shifts (Fig. 8 (d)-(e)). Also, Fig. 8(f) shows the behavior of HY when the target is completely occluded by the legs. HY maintains the occlusion recovery properties of PF, as the spread of the particles is sufficient to recover the target when it reappears.

Figure 7 also shows that MS-MP, PF-C-MP and HY-MP (the algorithms with the proposed multi-part representation) have better performance than their counterparts (MS, PF-C and HY) in terms of average lost tracks (λ_R), shape ($\bar{\mathcal{D}}_R$) and centroid ($\bar{\eta}_R$) errors. Overall, the best algorithm is HY-MP. In particular, MP improves the tracking performance when a target has a non-uniform color distribution (Table 2). For example, HY outperforms HY-MP only on the small and uniformly colored table tennis ball (*S2*). Also, due to the lower sampling of the sub-parts, the multi-part estimation of the MS vector becomes more unstable than that based on a single color histogram. This problem can be solved by analyzing the target and using a target-size threshold under which the single histogram representation should be used. A few results of Table 2 need further discussion: in *S1* (3D case) HY and HY-MP have similar λ_R, but HY-MP improves by around 20% in terms of $\bar{\mathcal{D}}_R$ and $\bar{\eta}_R$; in the easier sequences *S5* and *S6*, the track is never lost, but better performance is achieved again on shape and centroid position estimation. A visual comparison is shown in Fig. 9 and Fig. 10. Unlike HY, HY-MP is not attracted to false targets with similar color properties: the spatial information introduced in the model avoids a lost track (Fig. 9, third row) and improves the overall quality (Fig. 9, fourth row). HY using a single color histogram generates a wrong orientation and size estimation (Fig. 10, top) as the target and the background have similar colors, and the representation is not able to distinguish correctly the face. The spatial information in HY-MP solves this problem (Fig. 10, bottom). Moreover, when a target is partially occluded (Fig. 10 (a)) the spatial information improves the final estimate.

To analyze the algorithms performance with *highly maneuverable targets*, Fig. 11 shows the tracking results while varying σ_x and σ_y. As a reference, the results of MS and MS-MP are also presented. For all the values of $\sigma_x = \sigma_y$, HY outperforms PF-C and MS in terms of the number of lost track frames. Likewise when we compare HY-MP with PF-C-MP and MS-MP. As the target

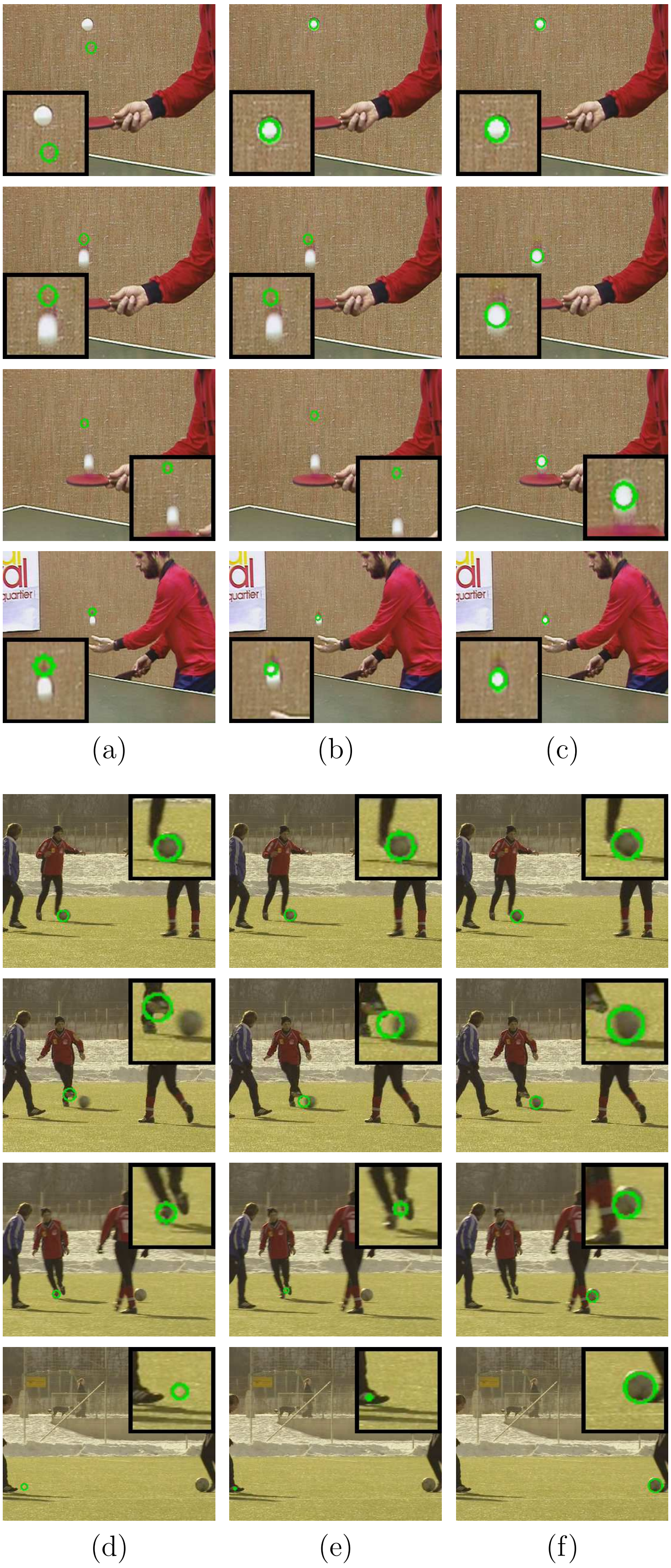


Fig. 8. Comparison of tracking performance. (a)-(c) *S2* (frames 3, 8, 25, 51), (d)-(f) *S3* (frames 1, 9, 18 and 52). Left column: MS; central column: PF-C; right column: HY.

Table 2
Comparison of tracking performance between the proposed algorithm (HY), Mean Shift (MS) and CONDENSATION (PF-C) for sequences with decreasing complexity (from top to bottom). Also, the multi-part representation (MP) is compared with the classic one based on a single color histogram (SH). Bold indicates the best result for the corresponding performance measure. Due to the deterministic nature of MS, standard deviation on the results is presented for PF-C and HY only.

			3D						**5D**			
			MS		**PF-C**		**HY**		**PF-C**		**HY**	
			SH	MP	SH	MP	SH	MP	SH	MP	SH	MP
HIGHLY MANEUVERING	**S1**	λ_R	.46	.44	.30	.25	**.02**	.03	.33	.29	.05	**.03**
		$\sigma(\lambda_R)$			.18	.06	.01	.01	.16	.11	.10	.02
		$\bar{\mathcal{D}}_R$	.33	.25	.40	.35	.29	**.23**	.43	.36	.29	**.23**
		$\sigma(\bar{\mathcal{D}}_R)$			.03	.02	.01	.00	.03	.03	.01	.00
		$\bar{\eta}_R$	.30	.24	.35	.32	.30	**.23**	.36	.32	.30	**.23**
		$\sigma(\bar{\eta}_R)$			.02	.01	.00	.00	.02	.02	.01	.00
	S2	λ_R	.84	.68	.40	.43	**.10**	.14	.37	.45	**.08**	.09
		$\sigma(\lambda_R)$			.07	.08	.06	.10	.08	.10	.02	.06
		$\bar{\mathcal{D}}_R$	.53	.24	.36	.31	**.15**	**.15**	.31	.33	**.14**	.15
		$\sigma(\bar{\mathcal{D}}_R)$			.06	.06	.02	.01	.04	.07	.01	.02
		$\bar{\eta}_R$	.37	.19	.28	.26	**.14**	.15	.27	.25	**.13**	.14
		$\sigma(\bar{\eta}_R)$			.03	.03	.02	.01	.02	.02	.01	.02
	S3	λ_R	.91	.91	.30	.11	.05	**.04**	.46	.23	**.04**	**.04**
		$\sigma(\lambda_R)$			.37	.21	.05	.02	.42	.34	.02	.01
		$\bar{\mathcal{D}}_R$	.15	.13	.30	.30	**.25**	**.25**	.25	.28	.25	**.22**
		$\sigma(\bar{\mathcal{D}}_R)$			.09	.05	.02	.01	.12	.08	.01	.01
		$\bar{\eta}_R$	.18	.16	**.17**	**.17**	.19	.19	**.15**	.17	.19	.17
		$\sigma(\bar{\eta}_R)$			.02	.02	.01	.01	.03	.03	.01	.01
CLUTTER	**S4**	λ_R	.26	.31	.17	.02	.16	**.01**	.23	.04	.19	**.02**
		$\sigma(\lambda_R)$			.04	.02	.06	.00	.05	.02	.04	.02
		$\bar{\mathcal{D}}_R$	.28	.22	.26	**.20**	.25	**.20**	.31	.24	.31	**.24**
		$\sigma(\bar{\mathcal{D}}_R)$			.01	.01	.01	.00	.02	.01	.02	.00
		$\bar{\eta}_R$	.32	.24	.24	**.15**	.23	**.15**	.29	.17	.30	**.16**
		$\sigma(\bar{\eta}_R)$			.01	.00	.01	.00	.02	.01	.02	.02
	S5	λ_R	.04	.01	**.01**	**.01**	**.01**	**.01**	**.01**	**.01**	.02	**.01**
		$\sigma(\lambda_R)$			.00	.00	.00	.00	.00	.00	.01	.00
		$\bar{\mathcal{D}}_R$	.25	.18	.20	0.18	.20	**.17**	.28	.23	.24	**.19**
		$\sigma(\bar{\mathcal{D}}_R)$			.00	.00	.00	.00	.01	.01	.01	.01
		$\bar{\eta}_R$	.22	.16	.18	0.17	.19	**.16**	.19	**.17**	.19	**.17**
		$\sigma(\bar{\eta}_R)$			.00	.00	.00	.00	.00	.00	.00	.00
EASY	**S6**	λ_R	**.00**	**.00**	**.00**	.02	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**
		$\sigma(\lambda_R)$			.00	.00	.00	.00	.00	.01	.00	.00
		$\bar{\mathcal{D}}_R$	.26	.25	.29	.29	.28	**.25**	.31	.32	.30	**.29**
		$\sigma(\bar{\mathcal{D}}_R)$			.03	.00	.01	.00	.03	.02	.01	.00
		$\bar{\eta}_R$	.27	.20	.20	**.13**	.21	.15	.21	**.13**	.21	**.14**
		$\sigma(\bar{\eta}_R)$			.00	.00	.00	.00	.00	.02	.00	.00

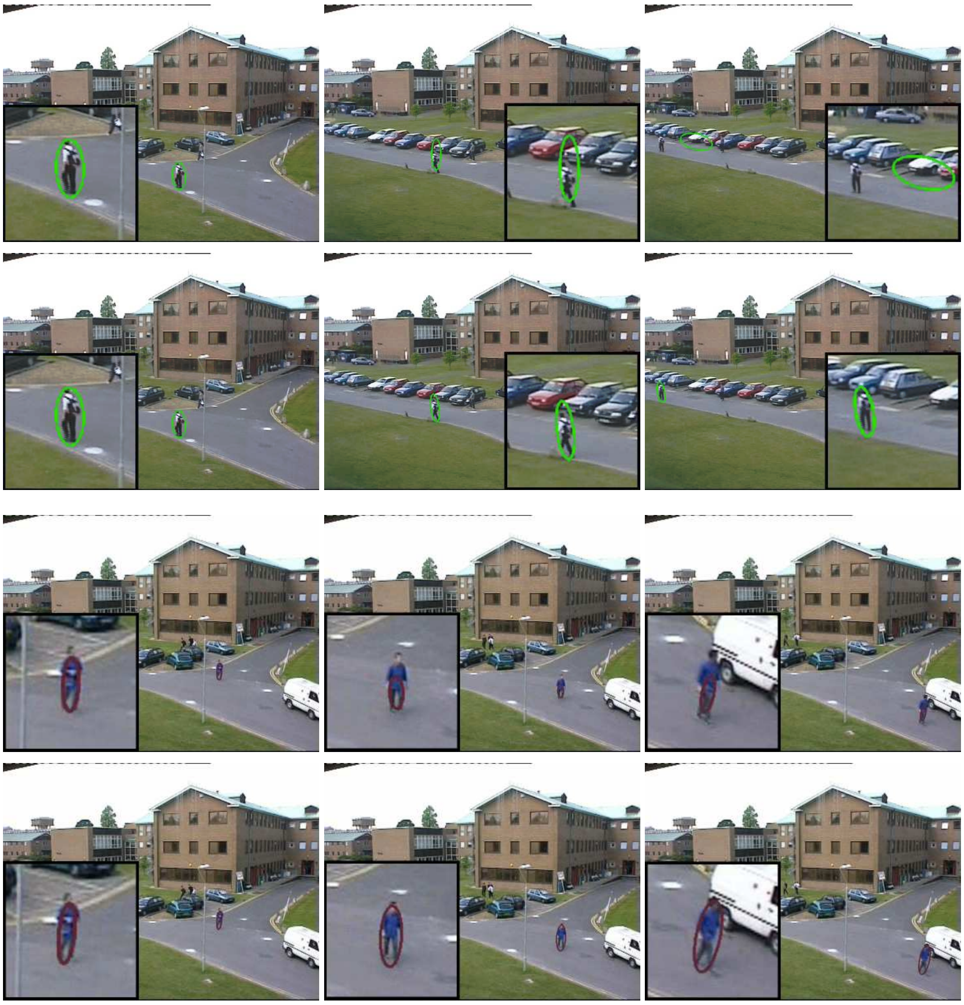


Fig. 9. Sample tracking results for the test scenario *S4* using different target representations: HY (first and third row) and HY-MP (second and fourth row). First and second row, *Camera 1, training*, frames 1520, 1625 and 1758; third and fourth row, *Camera 1, testing*, frames 1108, 1174 and 1235.

performs abrupt and fast movements, the sampling based on the predicted prior (PF-C and PF-C-MP) is inefficient, whereas particles concentrated on the peaks of the likelihood (HY and HY-MP) produce a better approximation. As a matter of fact the MS procedure increases the robustness of the algorithm to inappropriate parameter settings. Also, thanks to a higher distinctiveness, the MP representation achieves better performance than the traditional single histogram when the noise parameters are large and an error due to clutter is more probable.

To test the *robustness* of the algorithms we run the tracker on several temporal subsampled versions of the sequence *S5*. This test simulates possible frame losses in the video acquisition phase or an implementation of the algorithm embedded in a platform with limited computational resources (lower frame rate). The results (Fig. 12) show that HY and HY-MP are less affected than

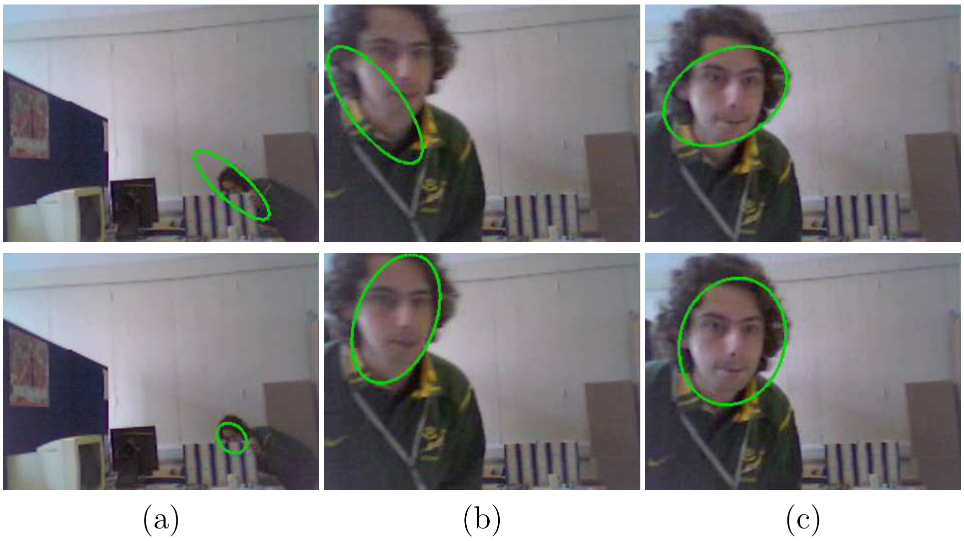


Fig. 10. Sample tracking results for the test sequence *S5* (frames 715, 819, and 864) using different target representations: HY (top) and HY-MP (bottom).

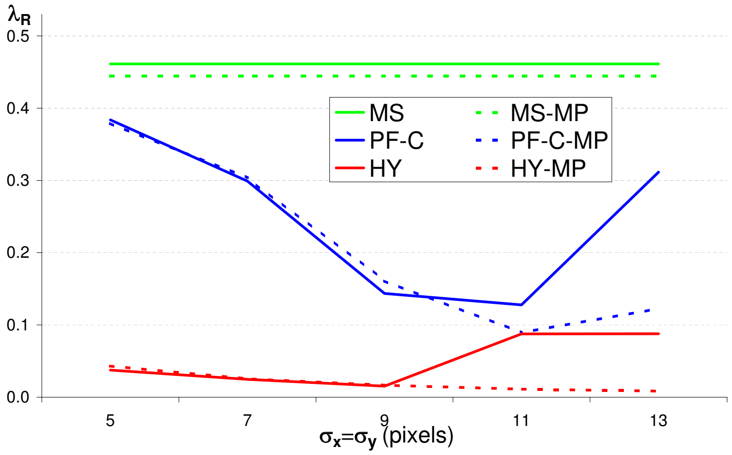


Fig. 11. Comparison of tracking results when varying the displacement random noise ($\sigma_x = \sigma_y$) of the motion model for the sequence *S1* (3D state space). The results of MS and MS-MP are shown as a reference. For all values of σ_x HY outperforms PF. Also, for high σ_x the proposed representation MP improves the result.

the other algorithms by a frame rate drop. HY and HY-MP perform similarly in terms of λ_R. However, on this type of test, HY-MP is always 10% to 15% better than HY in terms of $\sigma(\bar{\mathcal{D}}_R)$ and $\sigma(\bar{\eta}_R)$.

To evaluate *sensitivity to initialization*, we tested the trackers by adding noise to the initial centroid $\mathbf{y}_I$ (Fig. 13, left). We normalized the Gaussian noise with respect to the target size by multiplying its standard deviation σ_I by the length of the initial ellipse minor axis. For a fair comparison we used sequence *S5*, where the trackers perform similarly at zero noise level. For each algorithm we computed the average lost track ratio λ_R over the same 100 random initializations for different levels of initialization noise (Fig. 13, right).

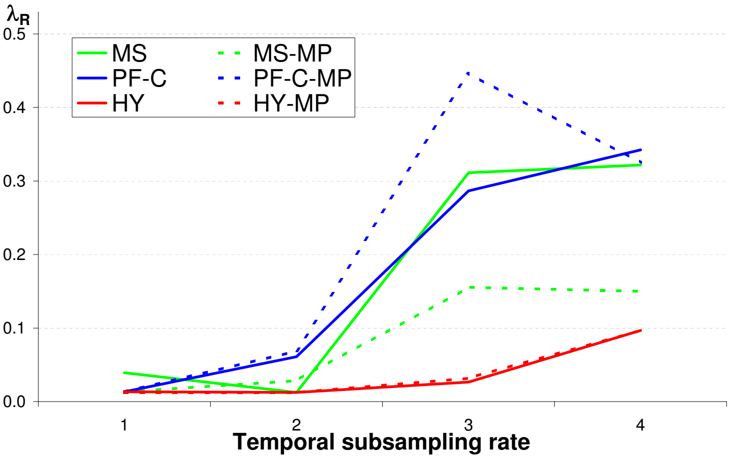


Fig. 12. Comparison of tracking results when varying the temporal subsampling rate of the input sequence *S5*. As the subsampling rate increases the movement of the object becomes less predictable; HY and HY-PF have more stable results than PF-C and MS, and achieve a lower λ_R (lost track ratio).

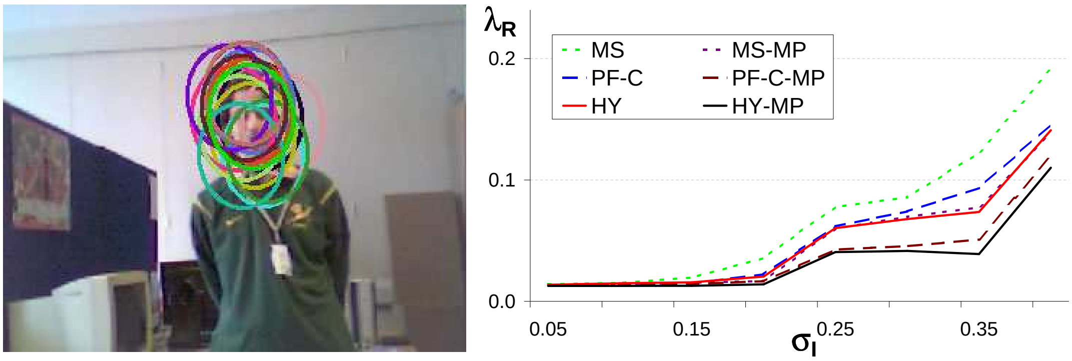


Fig. 13. Comparison of tracking results when the initialization (centroid) is affected by noise (*S5*). Left: sample random initializations ($\sigma_I = 0.4$). Right: average lost track ratios, λ_R, plotted against σ_I. As the noise increases, PF-C and HY show similar performance and are both more stable than MS. Also, the multi-part representation (MP) has lower λ_R (more stable) than the standard color histograms.

The curves are plotted up to $\sigma_I = 0.4$; for larger values several initial ellipses have more than 50% of their area outside the actual target. The results show that the trackers are fairly insensitive to initialization when $\sigma_I < 0.2$. For larger σ_I, HY and PF-C perform similarly and are less sensitive then MS due to multiple tracking hypotheses. Also, Fig. 13 shows that MP is more stable than the classical single histogram, especially with MS. In fact, unlike the whole ellipse, some sub-parts used in MP still fully overlap the actual target after the centroid shift and therefore their histograms are not biased by pixels belonging to the background.

As for the *computational cost*, we measured the running time of the C++ implementation of the algorithms (Pentium 4, 3GHz, with 512MB of RAM). For a fair comparison, we used *S5*, where none of the trackers fails. Also, the computational cost is linearly dependent with the number of pixels inside the target area (Eq. (1)); consequently, the results on *S5*, the largest target (see Table 1), represent an upper bound over the dataset.

MS, PF-C and HY, using the single histogram representation on the 3D state space, take 1.1ms, 6.9ms and 5.1ms per frame, respectively[3]; whereas using the MP representation, the values are 2.4ms (MS-MP), 15.6ms (PF-C-MP) and 15.9ms (HY-MP). When the target is represented with a single color histogram, HY is 6.3 times slower than MS, and 15% faster than PF-C. Compared to PF-C, the higher computational cost per particle of HY (due to MS iterations) is compensated by a higher sampling efficiency. In fact, HY outperforms PF-C using only 25% of the samples. When the MP representation is used, the computational cost of HY-MP is higher than that of MS-MP and comparable with that of PF-C-MP. MP makes HY proportionally slower with respect to PF-C, due to the larger number of histogram bins and consequently to the grater number of evaluations of Eq. (15). HY-MP, MS-MP and PF-C-MP are two to three times slower than their single histogram counterparts. This can be explained by analyzing our implementation. To compute a single color histogram, the image pixels inside the ellipse are scanned sequentially (row by row). Then, for each pixel, the bottleneck is to calculate the index of the bin to be incremented based on the pixel color (i.e., the evaluation of $b(.)$ in Eq. (1)) and to access its memory location. These operations account for about 45% of the overall computational cost[4]. In the multi-part case, each pixel belongs to three parts: the whole ellipse, one of the four sectors and inner or outer ellipse. Therefore we increment three bins for each pixel. We align in memory the histograms of the seven parts in such a way that, once we know the position of the first bin to increment, the memory location of the remaining two can be recovered by simply incrementing the pointer. This process adds two extra memory accesses to the computations.

6 Conclusions

We presented a tracking algorithm that uses a target representation based on multiple semi-overlapping color histograms and effectively combines Particle Filtering and Mean Shift in a principled way. The proposed target representation is general and takes into account target rotations and anisotropic scale changes, and achieves more accurate results in predicting target orientation and size than the single histogram and the multiple non-overlapping ones.

[3] The computational time measures were obtained using the C function `clock()` and does not include frame acquisition and decoding.

[4] This figure was obtained by profiling the C++ implementation of the algorithm.

The target partition is independent from the target class and incorporates global and local target information in a single model. The partition maintains the flexibility and robustness of the color histogram, while improving the performance of the traditional single histogram based tracker. The extension of the MS procedure to multiple histograms allows us to use the new target representation with the proposed HY algorithm. HY overcomes the drawbacks of both MS and PF-C, and makes each particle independent and more flexible to local conditions. Each particle is driven by MS in the position state sub-space directly using the color information to approximate the gradient.

Experimental results show that the proposed hybrid algorithm is more reliable than MS and, as the particles drawn are more efficient, HY is more accurate than PF-C even when only 25% of the particles are used. The best performance improvements are achieved with fast moving objects. The results show also that HY can handle short occlusions by propagating multiple tracking hypotheses generated by Monte Carlo sampling.

As future work, we will investigate the treatment of longer occlusions by means of a predictive higher-order dynamic model. Also, an occlusion detection strategy to trigger the switch to a different observation model may be adopted [41]. As for the appearance model, temporal adaptation will be investigated to cope with large pose changes. To this extent, when a priori information on the target appearance is available, a solution to fuse this information in a multi-pose model will be studied.

A Appendix: proof of Mean Shift for multi-part target representations

We demonstrate that, for multi-part color histograms (Eq. (17)), the iterative step $\mathbf{y}_b - \mathbf{y}_a$ of the Mean Shift (MS) procedure is in the direction of the gradient of ρ_{MP}, with respect to $\mathbf{y}$ (Eq. (5)). The Taylor expansion of ρ_{MP}, computed around $\{r_j(\mathbf{x}_a, \mathbf{z}_t)\}_{j=1,\dots,N}$, is

$$\begin{aligned}\rho_{MP}\left[\mathbf{r}(\mathbf{x},\mathbf{z}_t),\mathbf{o}\right] \approx & \frac{1}{2N}\left[\sum_{j=1}^{N}\sum_{u=1}^{U_j}\sqrt{\rho\left[r_{j,u}(\mathbf{x}_a,\mathbf{z}_t),o_{j,u}\right]}+\right.\\ & \left.+\sum_{j=1}^{N}\sum_{u=1}^{U_j} r_{j,u}(\mathbf{x},\mathbf{z}_t)\sqrt{\frac{o_{j,u}}{r_{j,u}(\mathbf{x}_a,\mathbf{z}_t)}}\right].\end{aligned} \tag{A.1}$$

By substituting Eq. (1) into Eq. (A.1), we obtain

$$
\begin{aligned}
\rho_{MP}\left[\mathbf{r}(\mathbf{x},\mathbf{z}_t),\mathbf{o}\right] \approx & \frac{1}{2N}\left[\sum_{j=1}^{N}\sum_{u=1}^{U_j}\sqrt{\rho\left[r_{j,u}(\mathbf{x}_a,\mathbf{z}_t),o_{j,u}\right]}+\right. \\
& \left.+\sum_{j=1}^{N}C_j(\mathbf{x})\sum_{i=1}^{n_j(\mathbf{x})}w_{j,i}k\left(\|A(\mathbf{s})\left(\mathbf{y}-\mathbf{w}_{j,i}\right)\|^2\right)\right].
\end{aligned}
\tag{A.2}
$$

The first term on the right hand side of Eq. (A.2) is independent from $\mathbf{y}$. Hence, to maximize ρ_{MP}, we have to maximize the second term on the right hand side. This second term is the kernel density estimate in $\mathbf{y}$, computed over the pixel positions $\mathbf{w}_{j,i}$, and weighted by $w_{j,i}$. This motivates the use of MS, which is a well-known solution to find the modes of a density function [42]. By computing the derivative of the Taylor expansion of ρ_{MP} with respect to $\mathbf{y}$ yields to

$$
\begin{aligned}
\frac{\partial\rho_{MP}\left[\mathbf{r}(\mathbf{x},\mathbf{z}_t),\mathbf{o}\right]}{\partial\mathbf{y}} \approx & \frac{1}{h_1^2N}\sum_{j=1}^{N}C_j(\mathbf{x})\,\cdot \\
& \cdot\sum_{i=1}^{n_j(\mathbf{x})}w_{j,i}g\left(\|A(\mathbf{s})\left(\mathbf{y}-\mathbf{w}_{j,i}\right)\|^2\right)B(\mathbf{s})\left(\mathbf{w}_{j,i}-\mathbf{y}\right),
\end{aligned}
\tag{A.3}
$$

where $g(x) = -k'(x)$ as in the single histogram case. Then

$$
\begin{aligned}
\frac{\partial\rho_{MP}\left[\mathbf{r}(\mathbf{x},\mathbf{z}_t),\mathbf{o}\right]}{\partial\mathbf{y}} \approx & \frac{1}{h_1^2N}B(\mathbf{s})\left[\sum_{j=1}^{N}C_j(\mathbf{x})\sum_{i=1}^{n_j(\mathbf{x})}\mathbf{w}_{j,i}w_{j,i}g(.)+\right. \\
& \left.-\mathbf{y}\sum_{j=1}^{N}C_j(\mathbf{x})\sum_{i=1}^{n_j(\mathbf{x})}w_{j,i}g(.)\right],
\end{aligned}
\tag{A.4}
$$

and, by multiplying outside and dividing inside the square brackets by the term $\sum_{j=1}^{N}C_j(\mathbf{x})\sum_{i=1}^{n_j(\mathbf{x})}w_{j,i}g(.)$, we obtain

$$
\begin{aligned}
\frac{\partial\rho_{MP}\left[\mathbf{r}(\mathbf{x},\mathbf{z}_t),\mathbf{o}\right]}{\partial\mathbf{y}} \approx & \frac{1}{h_1^2N}\left(\sum_{j=1}^{N}C_j(\mathbf{x})\sum_{i=1}^{n_j(\mathbf{x})}w_{j,i}g(.)\right)\cdot \\
& \cdot B(\mathbf{s})\left[\frac{\sum_{j=1}^{N}C_j(\mathbf{x})\sum_{i=1}^{n_j(\mathbf{x})}\mathbf{w}_{j,i}w_{j,i}g(.)}{\sum_{j=1}^{N}C_j(\mathbf{x})\sum_{i=1}^{n_j(\mathbf{x})}w_{j,i}g(.)}-\mathbf{y}\right],
\end{aligned}
\tag{A.5}
$$

where the argument of $g(.)$ is omitted for improved readability. By comparing this result with the classic sample-based MS formulation [43], we note that the term inside the square brackets multiplied by the matrix $B(\mathbf{s})$ is the MS vector $V(\mathbf{x})$, that is

$$V(\mathbf{x}) = B(\mathbf{s}) \left[\frac{\sum_{j=1}^{N} C_j(\mathbf{x}) \sum_{i=1}^{n_j(\mathbf{x})} \mathbf{w}_{j,i} w_{j,i} g(.)}{\sum_{j=1}^{N} C_j(\mathbf{x}) \sum_{i=1}^{n_j(\mathbf{x})} w_{j,i} g(.)} - \mathbf{y} \right],$$

where the contribution of each part is weighed by the normalization factor $C_j(\mathbf{x})$. An important difference with respect to [43] is the introduction of the matrix $B(\mathbf{s})$ to account for kernel rotation and anisotropic scaling. Given the initial centroid $\mathbf{y}_a$, the mean-shifted centroid position $\mathbf{y}_b$ for multiple histograms is derived by adding to $\mathbf{y}_a$ the MS vector evaluated in $\mathbf{x}_a$, that is

$$\begin{aligned} \mathbf{y}_b =& V(\mathbf{x}_a) + \mathbf{y}_a \\ =& B(\mathbf{s}_a) \left(\frac{\sum_{j=1}^{N} C_j(\mathbf{x}_a) \sum_{i=1}^{n_j(\mathbf{x}_a)} \mathbf{w}_{j,i} w_{j,i} g\left(\|A(\mathbf{s}_a)(\mathbf{y}_a - \mathbf{w}_{j,i})\|^2 \right)}{\sum_{j=1}^{N} C_j(\mathbf{x}_a) \sum_{i=1}^{n_j(\mathbf{x}_a)} w_{j,i} g\left(\|A(\mathbf{s}_a)(\mathbf{y}_a - \mathbf{w}_{j,i})\|^2 \right)} - \mathbf{y}_a \right) + \mathbf{y}_a. \end{aligned} \tag{A.6}$$

When the target is represented by one histogram only (i.e., $N = 1$), Eq. (A.6) reduces to Eq. (16). Also, the original MS formulation ([7]) can be seen as a special case of Eq. (A.6) with $h_1 = h_2$ and $\theta = 0$. Finally, if we rewrite $V(\mathbf{x})$ as a function of the gradient we obtain

$$V(\mathbf{x}) = \frac{h_1^2 N}{\sum_{j=1}^{N} C_j(\mathbf{x}) \sum_{i=1}^{n_j(\mathbf{x})} w_{j,i} g(\|A(\mathbf{s})(\mathbf{y} - \mathbf{w}_{j,i})\|^2)} \frac{\partial \rho_{MP}[\mathbf{r}(\mathbf{x}, \mathbf{z}_t), \mathbf{o}]}{\partial \mathbf{y}}. \tag{A.7}$$

This demonstrates our claim that, for multi-part color histograms, the MS vector $V(\mathbf{x})$, and consequently the MS step $V(\mathbf{x}_a) = \mathbf{y}_b - \mathbf{y}_a$, are in the direction of the gradient of ρ_{MP}.

Note that $V(\mathbf{x})$ is rescaled with respect to the gradient magnitude by the weighted kernel density estimate in $\mathbf{y}$ with kernel $g(.)$. In practice, MS is an adaptive gradient estimate that gives large response when the candidate in $\mathbf{y}$ poorly matches the model [7].

References

[1] B. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: Proc. of International Joint Conf. on Artificial Intelligence, 1981, pp. 674–679.

[2] S. K. Zhou, R. Chellappa, B. Moghaddam, Visual tracking and recognition using appearance-adaptive models in particle filters, IEEE Trans. Image Processing 13 (11) (2004) 1491–1506.

[3] I. Matthews, T. Ishikawa, S. Baker, The template update problem, IEEE Trans. Pattern Analysis Machine Intell. 26 (6) (2004) 810–815.

[4] H. T. Nguyen, A. W. M. Smeulders, Fast occluded object tracking by a robust appearance filter., IEEE Trans. Pattern Analysis Machine Intell. 26 (8) (2004) 1099–1104.

[5] A. Jepson, D. Fleet, T. El-Maraghi, Robust online appearance models for visual tracking, in: Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition, Vol. 1, Kauai, USA, 2001, pp. 415–422.

[6] M. Black, A. Jepson, Eigen–tracking: Robust matching and tracking of articulated objects using a view–based representation, International Journal on Computer Vision 36 (2) (1998) 63–84.

[7] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Trans. Pattern Analysis Machine Intell. 25 (5) (2003) 564–577.

[8] P. Perez, C. Hue, J. Vermaak, M. Gangnet, Color-based probabilistic tracking, in: Proc. of the European Conf. on Computer Vision, Vol. 1, Copenhagen, DK, 2002, pp. 661–675.

[9] S. Birchfield, Elliptical head tracking using intensity gradients and color histograms, in: Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition, Santa Barbara, USA, 1998, pp. 232–237.

[10] M. Isard, J. MacCormick, Bramble: A bayesian multiple-blob tracker., in: Proc. of International Conf. on Computer Vision, Vancouver, CAN, 2001, pp. 34–41.

[11] S. Birchfield, S. Rangarajan, Spatiograms versus histograms for region-based tracking., in: Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition, San Diego, USA, 2005, pp. 1158–1163.

[12] K. Okuma, A. Taleghani, N. D. Freitas, J. Little, D. G. Lowe, A boosted particle filter: Multitarget detection and tracking, in: Proc. of the European Conf. on Computer Vision, Prague, CZ, 2004, pp. 28–39.

[13] E. Maggio, A. Cavallaro, Multi-part target representation for colour tracking, in: Proc. of IEEE International Conf. on Image Processing, Genoa, IT, 2005, pp. 729–732.

[14] T. Liu, H. Chen, Real-time tracking using trust-region methods, IEEE Trans. Pattern Analysis Machine Intell. 26 (3) (2004) 397–402.

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

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

[17] K. Nummiaro, E. Koller-Meier, L. V. Gool, An adaptive color-based particle filter, Image and Vision Computing 21 (1) (2002) 99–110.

[18] M. Isard, A. Blake, CONDENSATION – conditional density propagation for visual tracking, International Journal on Computer Vision 29 (1) (1998) 5–28.

[19] R. van der Merwe, A. Doucet, N. de Freitas, E. Wan, The unscented particle filter, Tech. Rep. CUED/F-INFENG/TR380, Cambridge University, Engineering Department (Aug. 2000).
URL `http://cslu.cse.ogi.edu/publications/ps/merwe00.pdf`

[20] K. Choo, D. Fleet, People tracking using hybrid Monte Carlo filtering., in: Proc. of International Conf. on Computer Vision, Vol. 2, Vancouver, CAN, 2001, pp. 321–328.

[21] J. Sullivan, J. Rittscher, Guiding random particles by deterministic search., in: Proc. of International Conf. on Computer Vision, Vol. 1, Vancouver, CAN, 2001, pp. 323–330.

[22] C. Chang, R. Ansari, Kernel particle filter: iterative sampling for efficient visual tracking, in: Proc. of IEEE International Conf. on Image Processing, Vol. 3, Barcelona, SP, 2003, pp. III–977–80.

[23] B. Han, D. Comaniciu, Y. Zhu, L. Davis, Incremental density approximation and kernel-based bayesian filtering for object tracking, in: Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition, Vol. 1, Washington, USA, 2004, pp. 638–644.

[24] C. Sminchisescu, B. Triggs, Covariance scaled sampling for monocular 3d body tracking, in: Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition, Vol. 1, Kauai, USA, 2001, pp. 447–454.

[25] C. Sminchisescu, B. Triggs, Hyperdynamics importance sampling, in: Proc. of the European Conf. on Computer Vision, Vol. 1, Copenhagen, DK, 2002, pp. 769–783.

[26] J. Deutscher, A. Blake, I. Reid, Articulated body motion capture by annealed particle filtering, in: Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition, Vol. 2, Head Island, USA, 2000, pp. 126–133.

[27] P. Li, T. Zhang, A. E. C. Pece, Visual contour tracking based on particle filters, Image and Vision Computing 21 (1) (2003) 111–123.

[28] C. Shan, Y. Wei, T. Tan, F. Ojardias, Real time hand tracking by combining particle filtering and mean shift, in: Proc. of IEEE International Conf. on Automatic Face and Gesture Recognition, Southampton, UK, 2004, pp. 669–674.

[29] E. Maggio, A. Cavallaro, Hybrid particle filter and mean shift tracker with adaptive transition model, in: Proc. of IEEE International Conf. on Acoustics, Speech, and Signal Processing, Vol. 2, Philadelphia, USA, 2005, pp. 221–224.

[30] T. Kailath, The divergence and Bhattacharyya distance measures in signal selection, IEEE Trans. Commun. Technol. 15 (1967) 52–60.

[31] D. Comaniciu, P. Meer, Mean shift: A robust approach toward feature space analysis, IEEE Trans. Pattern Analysis Machine Intell. 24 (5) (2002) 603–619.

[32] B. Silverman, Density Estimation for Statistics and Data Analysis, Chapman and Hall, London, 1986.

[33] G. Pingali, J. Segen, Performance evaluation of people tracking systems, in: Proc. of the IEEE Workshop on Applications of Computer Vision, Sarasota, USA, 1996, pp. 33–38.

[34] V. Mariano, M. Junghye, P. Jin-Hyeong, R. Kasturi, D. Mihalcik, L. Huiping, D. Doermann, T. Drayer, Performance evaluation of object detection algorithms, in: Proc. of IEEE Conf. on Pattern Recognition, Vol. 3, Quebec, CAN, 2002, pp. 965–969.

[35] C. Needham, R. Boyle, Performance evaluation metrics and statistics for positional tracker evaluation, in: Proc. of International Conf. on Computer Vision Systems, New York, USA, 2003, pp. 278–289.

[36] J. Black, T. Ellis, P. Rosin, A novel method for video tracking performance evaluation, in: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), Nice, FR, 2003, pp. 125–132.

[37] T. Schlogl, C. Beleznai, M. Winter, H. Bischof, Performance evaluation metrics for motion detection and tracking, in: Proc. of IEEE Conf. on Pattern Recognition, Vol. 4, Surrey, UK, 2004, pp. 519–522.

[38] D. Doermann, D. Mihalcik, Tools and techniques for video performance evaluation, in: Proc. of IEEE Conf. on Pattern Recognition, Vol. 4, Barcelona, SP, 2000, pp. 167–170.

[39] C. Jaynes, S. Webb, R. Steele, Q. Xiong, An open development environment for evaluation of video surveillance systems, in: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), Copenhagen, DK, 2002, pp. 32–39.

[40] R. Collins, X. Zhou, S. Teh, An open source tracking testbed and evaluation web site, in: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), Breckenridge, USA, 2005.

[41] Y. Wu, G. Hua, T. Yu, Switching observation models for contour tracking in clutter, in: Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition, Madison, USA, 2003, pp. 295–304.

[42] K. Fukunaga, L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Trans. Inform. Theory 21 (1) (1975) 32–40.

[43] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition, Vol. 2, Head Island, USA, 2000, pp. 142–149.