Continuously adaptive data fusion and model re-learning for particle filter tracking with multiple features

Xiao, Jingjing; Stolkin, Rustam; Oussalah, Mourad; Leonardis, Ales

DOI:
10.1109/JSEN.2016.2514704

License:
Other (please specify with Rights Statement)

Document Version
Peer reviewed version

Citation for published version (Harvard):
Xiao, J, Stolkin, R, Oussalah, M & Leonardis, A 2016, 'Continuously adaptive data fusion and model re-learning for particle filter tracking with multiple features', *IEEE Sensors Journal*, vol. 16, no. 8.
https://doi.org/10.1109/JSEN.2016.2514704

Continuously adaptive data fusion and model re-learning for particle filter tracking with multiple features

Jingjing Xiao[1*], Rustam Stolkin[2], Mourad Oussalah[1], Aleš Leonardis[3]
[1] School of Electronics, Electrical and Computer Engineering, University of Birmingham, B15 2TT, UK
[2]School of Mechanical Engineering, University of Birmingham, B15 2TT, UK
[3]School of Computer Science, University of Birmingham, B15 2TT, UK

***Abstract*—** This paper presents a new method for object tracking in a camera sensor with particle filters. The method enables multiple target and background models, spanning arbitrarily many features or imaging modalities, to be adaptively fused to provide optimal discriminating ability against changing backgrounds, which may present varying degrees of clutter and camouflage for different kinds of features at different times. Furthermore, we show how to continuously and robustly relearn all models for all feature modalities online during tracking, for targets whose appearance may be continually changing. Both the data fusion weightings and model re-learning parameters are robustly adapted at each frame, by extracting contextual information to inform saliency assessments of each part of each model. Additionally, we propose a two-step estimation method for improving robustness, by preventing excessive drifting of particles during tracking past challenging, cluttered background scenes. We demonstrate the method by implementing a version of the tracker which combines both shape and colour models, and testing it on a publicly available bench-mark data set. Results suggest that the proposed method outperforms a number of well-known state-of-the-art trackers from the literature.

***Keywords*—** visual object tracking, particle filter, colour histogram, HOG feature, data fusion, online model learning

I. INTRODUCTION

A. Robust approaches to visual object tracking

After several decades of effort, tracking an object in the camera sensor remains an open research problem [21]. Two main streams of research can be distinguished. The first one explores the use of increasingly sophisticated visual features and representational models of the tracked target, while the second espouses spatiotemporal filtering and searching methods. Many different cues can be utilized for target representation, e.g. colour, shape, silhouette and others [22]. It is increasingly believed that robust tracking cannot be achieved with a single feature and recent work has explored methods for combining information from multiple features in various robust ways [13, 23, 25, 32, 37]. For spatiotemporal filtering, several approaches have emerged as popular choices over the past 15 years (e.g. meanshift tracking [9], Kalman filtering [35], particle filtering [20]). Particle filters are a simple but powerful recursive tracking model which have demonstrated great potential for handling multi-modal problems of general non-linear and non-Gaussian systems. The particle filter was first introduced for tracking by Isard and Blake [26], who showed how to use it as an underlying spatio-temporal filter for contour tracking. It appears that both Nummiaro [18, 20], and Perez [29] independently (in 2002) co-invented a method, that has now become widespread for using particle filters with the colour histogram target model, first proposed by Ennesser and Medioni as early as 1995 [7], but later widely popularized by the work of Comaniciu et al. (2000-2003) in the mean-shift tracking literature [2, 4]. Although state-of-the-art in the early 2000s, the above methods were starting to show their age by the late 2000s, and became superceded by more sophisticated and complex methods which demonstrated significantly improved performance [3, 12, 36]. In particular, the recent work [13, 23], showed how colour, shape and motion cues could be combined to achieve state-of-the-art tracking performance. However, these methods relied on complex and intricate two-layer local and global model combinations, engendering correspondingly complex and expensive algorithmic machinery for tracking and model updating.

In this paper, we revisit the simple and convenient architecture of the histogram-based particle filter. We show how it can be modified to achieve competitive performance against the most sophisticated modern methods, by the following three enhancements (a shorter version [38] was published in IPTA):

1) A continuously adaptive data fusion method for optimally combining multiple features.

2) A more robust method for continuous re-learning of targets which change their appearance, while avoiding the accidental relearning of background features into the target models.

3) A two-step estimation method to prevent excessive particle drifting.

B. Particle filtering with fusion of multiple feature modalities

The original particle filter based trackers [18, 26, 29] used only a single feature for modeling the target. Increasingly sophisticated performance evaluation efforts within the vision community [21, 24] are revealing the limitations of such approaches, motivating the integration of more features. This paper asserts that the two fundamental issues with combining multiple features during dynamic tracking are: 1) how to dynamically determine which features are most discriminatory in changing scenes; 2) how to adaptively weight in favour of the most salient features while devaluing the contribution from poor features.

Simplistic approaches of merely multiplying the contributions of each feature modality, i.e. as a product of probabilities (or a "product of experts" [8]) will be prone to failure in scenes where one or more of the feature modalities is challenged by clutter of similar appearance to the target. This is for the simple reason that multiplication of feature weights can cause a false negative detection by a poorly performing feature. Perez et al. [30] suggested fusing arbitrary numbers of features by weighting and resampling particles according to the observations of each feature in succession. This method therefore assigns equal importance to all features and offers no method for adaptively weighting in favour of good features over bad features during tracking. Additionally, the work of [30] did not incorporate any methods for automatically updating or relearning the target model online, which is necessary for substantially changing scenes, changing views of the target, and highly deformable targets. Brasnett et al [31] proposed a scheme for weighting in favour of the best performing features and updating these weights adaptively at each new frame. The significance of each feature modality is weighted according to how well the target (reference) model for that feature matches the best current candidate target region of the image. This approach fulfills one of the requirements in our above discussion of saliency (good features should return high weights for target regions of the image), but ignores the other critical requirement (a discriminating feature must *also* return low weights for local

*Corresponding author: Jingjing Xiao, Email：shine636363@sina.com

background regions). Maggio et al. [6] proposed an alternative approach to evaluating feature saliency, based on examining the statistics of the spatial distribution of the particles themselves. Essentially, they propose a heuristic wherein those features whose weights suggest a tight clustering of particle positions (in image spatial coordinates) are considered more salient than those features associated with a broader spatial spread of particles. This method may also work satisfactorily under benign tracking conditions, but essentially throws away the benefits of the particle filter for tracking past cluttered backgrounds. Therefore, selecting features which reduce the distribution to a single concentrated cluster removes the robustifying effect of the particle filter, rendering its performance similar to that of a simple Kalman filter based on uni-modal distributions over target. Our recent work [25] showed how the popular histogram-based particle filter tracker of [20] could be extended to incorporate data fusion of several different features or imaging modalities, and proposed an adaptive method for optimally weighting the contributions from each feature online during tracking. However, for optimal performance, this method requires that each particle's local background region be separately re-modeled at each frame, and this rapidly becomes computationally expensive as the number of particles is scaled.

In this paper, we propose a new method for adaptive feature fusion. The new method uses the statistics of the distribution of particle weights themselves as a cue to feature saliency. The underlying principle is that particles inherently contain information about the background as well as the target. In simple terms, "bad" particles will encode background information, while "good" particles encode target information. Hence, we are able to extract information about the saliency of each feature by examining the spread of particle weightings suggested by each feature modality.

C. Continuous relearning of target models during tracking

Continuously relearning the reference target model is inherently dangerous. There will always be some noise and uncertainty in the estimated target location, which usually includes a significant number of background pixels. This degrades tracking performance further, leading to even more erroneous target relearning. Much of the particle filter literature follows the simple target update method proposed in [20], where a target histogram is updated every frame as a simple leaky linear combination of the previous model and current estimated status. Without additional methods for precise delineation of the target parts, such update methods are prone to failures. Later work [15] suggested a more sophisticated target updating scheme that utilizes a decision of minimum error over the whole particle distribution. Work in [17] proposed a Rao-Blackwellised Particle Filter (RBPF) for handling the uncertainties caused by illumination changes and brief periods of occlusion. Nevertheless, both [15] and [17] based target model updates solely on the information extracted from the estimated target region, where any inaccuracies in estimation are likely to lead to serious drifting problems over time.

In this paper, we enable stable and robust model updating by identifying those parts of the feature space, which are highly prominent in the current foreground while also being of very low prominence in the local background region.

D. Contributions of this work

This paper shows how simple histogram-based particle filter trackers can be robustified by proposing the following enhancements:

1) A continuously adaptive data fusion method for multiple features. We propose a new measure of feature saliency, which is both robust and also computationally cheap, derived from the standard deviation of the particles' weights in each feature modality. This is combined with more conventional measures of candidate region and reference model consistency, in order to generate an overall saliency metric which selects features that both provide high weightings for target-like pixels and also low weightings for local background-like pixels.
2) A robust method for continuous re-learning of targets which change their appearance, while avoiding accidental relearning of background features into the target models. We extract contextual information about the local background image statistics, and use this to determine which elements of the estimated current target region are most distinct from the background. Updating of the target reference model is then weighted in favour of the most distinct elements of the current target region.
3) A two-step estimation method for preventing excessive particle drifting. A first-round estimation is used to detect and re-initialise drifting particles. A second-round estimation, then re-samples from the modified set of particles, in order to more robustly re-estimate the new target location.

E. Layout of this paper

The basic framework of the Particle Filter is introduced in Section II. Our proposed method for adaptive fusion of multiple features is explained in Section III. A two-step estimation procedure is developed in Section IV. Section V explains the method for online model adaptation. Occlusion handling is presented in Section VI. Section VII describes our performance evaluation experiments. Section VIII provides concluding remarks.

II. Particle filter framework

At each time step, we represent the state X of the tracked target by a distribution, approximated by a weighted set of N particles, with associated weights:

$$X = \sum_{i=1}^{N} s^{(i)} \omega^{(i)} \quad (1)$$

where $s^{(i)}$ represents a candidate target state, referred to as the i^{th} particle, with associated weight $\omega^{(i)}$ such that $\sum_{i=1}^{N} \omega^{(i)} = 1$. As is common in much of the tracking literature, we model the target by a simple rectangular region, where $s^{(i)}$ includes the parameters for describing the position, scale and velocity of the corresponding rectangular bounding box:

$$s^{(i)} = [x \;\; y \;\; \dot{x} \;\; \dot{y} \;\; x_w \; y_h]^T \quad (2)$$

where x, y, $\dot{x}$, $\dot{y}$ are the position and velocity of the target in image coordinates, and x_w and y_h correspond to the width and height of the rectangular target region, centered at x, y, which constitutes the a bounding box.

The algorithm first propagates the particles according to a motion model, namely:

$$s_k = A s_{k-1} + v_{k-1} \quad (3)$$

where $v_k \sim G(0, R)$ is a zero mean Gaussian noise with variance-covariance matrix R. For simplicity, we utilize a first order motion model where A corresponds to constant velocity. Features are extracted from image data and used to evaluate the likelihood of each particle, according to measures of similarity between that particle's image region and the target model.

III. Adaptive online fusion of multiple target feature models

Our method can conveniently be applied to fuse data from any kind of target image features which can be expressed as a histogram model. For proof of principle, we here explain the method in terms of fusing colour histograms and HOG features [27] since this pair of features are known to be particularly complementary.

A. Colour histogram target model

For the i^{th} pixel, of an image region (e.g. bounding box), R, with colour, u, we use the function:

$$u_i \mapsto h(u_i) \quad (4)$$

*Corresponding author: Jingjing Xiao, Email：shine636363@sina.com

where h: $\mathbb{R}^2 \rightarrow \{1,2,\dots,M\}$ of bins in a colour histogram, according to its RGB colour value. The probability of a particular histogram bin, ζ, is then returned by:

$$H_{color}(\zeta) = \left\{\frac{1}{M}\sum_{i\in R}\delta[h(u_i^*) - \zeta]\right\}_{\zeta=1\dots m} \quad (5)$$

where δ is the Kronecker delta function. Notice that H_{color} is normalized so that: $\sum_{\zeta=1}^{m} H_{color}(\zeta) = 1$.

B. Histogram of oriented gradients target model

The histogram of gradients (HOG) feature [27] is employed in our work to encode target shape information. HOG features represent object shape within an image as a distribution of gradient intensities with respect to edge directions, shown in Fig. 1:

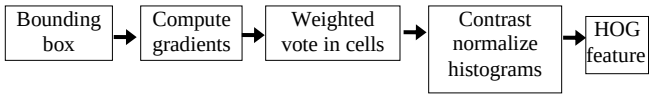

Figure 1. Procedure of extracting HOG feature

1) Gradient computation: in this step a 1-D centered, point discrete derivative mask in both the horizontal and the vertical directions, is employed. Specifically, this method requires filtering the colour or intensity data of the image with the following filter kernels:

$$[-1,0,1] \text{ and } [-1,0,1]^T \quad (6)$$

For each pixel, the norm and orientation are computed by:

$$norm(x,y) = \sqrt{px^2(x,y) + py^2(x,y)} \quad (7)$$

$$orient(x,y) = arc\ tan(py(x,y)/px(x,y)) \quad (8)$$

where $px(x,y)$ and $py(x,y)$ represent the horizontal and vertical gradient values, respectively.

2) Orientation binning: the image region of interest is divided up into rectangular cells. Each cell is associated with an edge-orientation histogram (9 bins per histogram in our tracker) and each pixel within the cell casts a weighted vote for a particular bin of the histogram. Hence, bin ζ for a cell histogram is computed as:

$$H_{cell}(\zeta) = \sum_{j=1}^{N_p} norm(x_p,y_p)\,\delta[orient'(x_p,y_p) - \zeta] \quad (9)$$

where δ is the Kronecker delta function and $orient'(x_p,y_p)$ is quantized orientation, computed from $orient(x,y)$. N_p is the number of pixels in each cell. We represent the set of sums of magnitude in gradient ζ for each cell as an N-orientation histogram:

$$H_{cell}^{all} = \{H_{cell}(1), H_{cell}(2), \dots, H_{cell}(N)\} \quad (10)$$

3) Descriptor bounding box: in our implementation, we divide a candidate bounding box into nine rectangular cells, each associated with a 9-bin edge orientation histogram. All nine cell histograms are now concatenated to make a single 81-dimensional feature vector H_{hog}. The cells share 50% overlap of their area. In other words, each pixel contributes to more than one cell to form the final histogram. To cope with the illumination and contrast changes, the gradient values of each cell are locally normalized, according to the gradient L2-norm:

$$H'_{hog}(\zeta) = H_{hog}(\zeta)/\sqrt{\left(\sum_{k=1}^{q*q*N} H_{hog}(k)^2\right) + \epsilon} \quad (11)$$

for $q * q$ cells ($q = 3$ in our tracker) and a regulation parameter $\epsilon = 0.01$. After normalization, the histogram for a particle's bounding box region becomes:

$$H_{cell}^{'all} = \{H'_{hog}(1), H'_{hog}(2), \dots, H'_{hog}(B * N)\} \quad (12)$$

where B is the number of cell regions ($B = q * q$) that are contained in the target bounding box region.

C. *Adaptively weighted feature combination*

When tracking is initialized (by designating a bounding box for the target in the first frame), a pair of target reference models are constructed for the target. At each frame, colour and HOG-feature histograms, H_{cand}^{color}, H_{cand}^{hog} are constructed from the bounding box region around each particle. Each of these particle histograms (both colour and HOG) can now be compared with their corresponding reference models using the Bhattacharya coefficient [1]:

$$D_{color}(H_{ref}^{color}, H_{cand}^{color}) = \sum_{h=1}^{M}\sqrt{H_{ref}^{color}(\zeta).H_{cand}^{color}(\zeta)} \quad (13)$$

$$D_{hog}(H_{ref}^{hog}, H_{cand}^{hog}) = \sum_{h=1}^{M}\sqrt{H_{ref}^{hog}(\zeta).H_{cand}^{hog}(\zeta)} \quad (14)$$

as similarity measures for colour and HOG features respectively. For each feature modality, it is now possible to compute a likelihood of the candidate region matching the target:

$${\omega_{color}}^{(i)} = \frac{1}{\sqrt{2\pi\sigma_{color}}}exp\{{D_{color}}^2/2{\sigma_{color}}^2\} \quad (15)$$

$${\omega_{hog}}^{(i)} = \frac{1}{\sqrt{2\pi\sigma_{hog}}}exp\left\{\frac{{D_{hog}}^2}{2{\sigma_{hog}}^2}\right\} \quad (16)$$

where, for the ith particle, likelihoods are represented as Gaussians with variances σ_{color}, σ_{hog}, which represent the noise associated with each feature modality. In our experiments, we determined the values of these parameters empirically, and find that values of 0.01 work well for both parameters. Future work will explore ways of learning and updating these parameters dynamically online. Note that both ${\omega_{color}}^{(i)}$ and ${\omega_{hog}}^{(i)}$ fulfill the normalization condition; namely, $\sum_{i=1}^{N}{\omega_{color}}^{(i)} = 1$, and $\sum_{i=1}^{N}{\omega_{hog}}^{(i)} = 1$.

We fuse the features by using a weighted combination of the coefficients, but use contextual information to continually update the weighting factor during tracking in a way that ensures optimal overall discriminating power of the combined feature model:

$$\omega^{(i)} = \mu_d{\omega_{color}}^{(i)} + (1 - \mu_d){\omega_{hog}}^{(i)} \quad (17)$$

where weighting factor μ_d takes values between 0 and 1. We now explain how to use contextual information to achieve online tuning of weighting factor μ_d to enable adaptation to changing scenes.

For online tuning of the weighting factor μ_d, we should design a performance metric which can quantify the discriminating ability of each feature. A key innovation of this paper is to note that background/foreground information is already encoded in the distribution of the particles themselves. Some particles will mostly encode information about target feature values, while other particles will encode information about background feature values.

We can exploit this property of the particle filter, by examining the distribution of the particle weights, ${\omega_{color}}^{(i)}$ and ${\omega_{hog}}^{(i)}$. A poor feature is one that does not discriminate between background and target regions. Therefore, such a feature will assign similar weights to particles lying on true target regions and particles lying on background regions. Hence we expect to see a small spread in the weight distributions, characterized by a small standard deviation of the weight values for that feature. In contrast, a highly discriminating feature is one that assigns high weights to particles lying on true target regions but low weights to particles lying on background regions. Therefore, assuming a good spatial spread of particles, a highly discriminative feature will exhibit a good spread of different particle weights ${\omega_{color}}^{(i)}$ and ${\omega_{hog}}^{(i)}$, which can be evaluated according to the standard deviation of the weight values σ_{colour}^{ω} and σ_{hog}^{ω}. In other words, the saliency of one particular feature should be proportional to its own standard deviation of weight values, while inversely to the standard deviation of other features, denoted as:

$$\text{saliency}_{colour} \propto \left(\frac{\sigma_{colour}^{\omega}}{\sigma_{hog}^{\omega}}\right),\ \text{saliency}_{hog} \propto \left(\frac{\sigma_{hog}^{\omega}}{\sigma_{colour}^{\omega}}\right) \quad (18)$$

Additionally we would also like features which return a high weight for particles lying on true target regions. Since particle filter tracking is a form of stochastic estimation, we can never know the true target location online during tracking, however it is most likely to be the region corresponding to that particle with the highest current weight, i.e. the highest value of Bhattacharyya coefficient between reference and candidate region histograms. Therefore, we use a second indicator of feature salience consisting of:

$$D_{colour}^{max} = max\{D_{colour}^{i}\}_{i=1\dots N} \quad (19)$$

$$D_{hog}^{max} = max\{D_{hog}^{i}\}_{i=1\dots N} \quad (20)$$

*Corresponding author: Jingjing Xiao, Email：shine636363@sina.com

where N represents the number of the particles. We can now combine all of these different saliency metrics into a single, overall feature weighting factor μ_d:

$$\mu_d = \frac{\sigma^{\omega}_{colour} D^{max}_{colour}}{\sigma^{\omega}_{hog} D^{max}_{hog} + \sigma^{\omega}_{colour} D^{max}_{colour}} \quad (21)$$

Note, that equation (21) is easily extendable in order to evaluate the relative saliency of an arbitrary number of different feature modalities. For F features, the appropriate weight, μ_f, for the f^{th} feature is found simply as:

$$\mu_f = \frac{\sigma^{\omega}_f D^{max}_f}{\sum_{f=1}^{F} \sigma^{\omega}_f D^{max}_f} \quad (22)$$

Thus, our method can easily extend to dynamic, online, adaptive weighting for data fusion of arbitrary numbers and combinations of feature and/or imaging modalities, by fusing them as a simple linear combination according to the above relative weights. This ensures the optimum discriminating capability of the feature set is exploited at each successive image frame. The overall scheme for online adaptive weighting of features is summarized in Fig. 2.

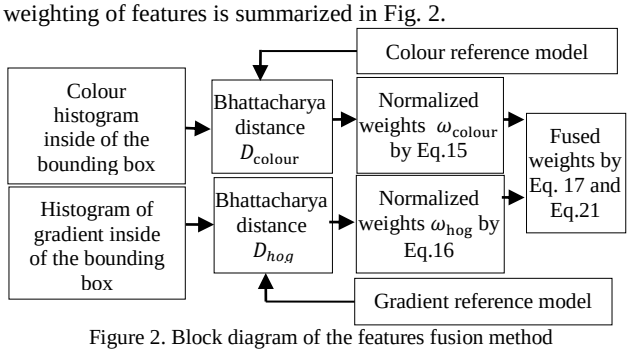

Figure 2. Block diagram of the features fusion method

IV. Two-step estimation for drift prevention

When tracking targets that move past cluttered backgrounds, where distracting parts of the background share common feature values with the target object, it is common for distant particles to be awarded high weights during the observation and update step of filtering. This can cause excessive drifting of particles, leading to eventual failure as the set of particles degenerates. It also can cause short term errors in estimating of the target location, shown in Fig. 3.

Figure 3. Failure modes in environmental clutter. In this case, the target walks from left to right across the scene, but is temporarily occluded by a distracting object, sharing similar feature values, which passes from right to left.

To address this problem, we propose a two-step estimation procedure which detects and replaces such drifting particles, prior to re-estimating the target location. After the conventional particle filter re-sampling and propagation steps, at the k^{th} time-step, we generate a first-round target estimation $\hat{X}_k$, as the weighted mean of all particle positions. We then calculate the set of distances d^i between $\hat{X}_k$ and the i^{th} particle, for all particles, and take the average of these to find the standard deviation $\bar{d}$ of particle distances wrt $\hat{X}_k$. We now detect drifting particles as those for which:

$$d^i > \lambda_d \bar{d} \quad (23)$$

where λ_d is a constant parameter which we set as 2 for the experiments described in this paper. Any particles which satisfy the constraint of equation 23 are removed, and replaced by new particles created at position $\hat{X}_k$. Next, we proceed in the usual way, by obtaining observation features for the new set of particles and use them to compute the Bhattacharyya metrics and particle weights (do Eq.13-22). Finally, we use Eq.1 to give the overall target position estimate for the current frame.

V Robust online model adaptation

In general, the appearance of the target will change with time, so that robust tracking can only be achieved by continuously updating the shape and colour target models. Previous work of ourselves and colleagues [13, 23] addressed this problem by representing the target as a coupled-layer combination of local (sets of small patches) and global (overall target region) models. During target re-learning, each model can provide stability by constraining the re-learning of the other model. These methods achieved state-of-the-art performance, but involved very complicated models engendering corresponding complexities in the resulting necessary tracking machinery. In contrast, we show how robust model adaptation can be enabled in a simple histogram-based particle filter, by making use of contextual information. Similar to the original simple update mechanism [20], we also use the feature histograms of the image region around the current target estimate to update the target reference models. However, we robustify the relearning, by only relearning those histogram bins which are far more prominent in the foreground region than the background region. This mechanism effectively prevents background feature values from being erroneously relearned into the target model. The method is conducted in three steps: background model extraction; determination of "relearning weights" for each histogram bin of each feature; updating the reference model with weighted elements from current foreground model.

A. *Online identification of local background models*

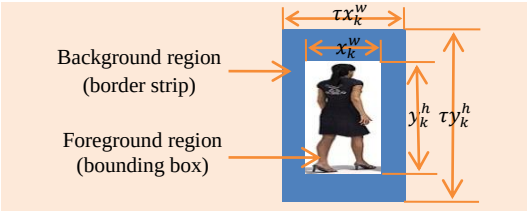

Figure 4. Foreground and background regions

We extract contextual information by enlarging the bounding box as shown in Fig.4. A local background region is defined as a border strip that surrounds the foreground region, where the background histogram extracted from. More specifically, we enlarge the bounding box around the current estimate of the target location, by a scaling factor τ (set 1.2 in the experiment), so that the size of the expanded region is given by:

$$A_{f+b} = \tau^2 A_f = \tau^2 x_k^w y_k^h \quad (24)$$

where x_k^w and y_k^h are the lengths of the bounding box in x and y axis, respectively, and A_f and A_{f+b} are the areas of the foreground region and the foreground+background regions respectively. Then, for each feature (colour and shape), we generate an appearance model (histogram) H_{f+b} for all pixels contained within the enlarged bounding box, which contains both foreground and background information. Then, for each feature modality, the bin ζ for the local background histogram can be calculated as:

$$H_b(\zeta) = \frac{A_{f+b} H_{f+b}(\zeta) - A_f H_f(\zeta)}{A_{f+b} - A_f} \quad (25)$$

B. *Identifying relearning weights for model bins*

After identifying the appearance model of the background according to (25), we use it to evaluate the relative prominence of each bin of each feature model in the foreground and background. For each feature modality, we define a "relearning weight" for each histogram bin u as:

$$c_u = 1 - e^{-\lambda_c \left(H_f(\zeta) / H_b(\zeta) \right)} \quad (26)$$

where λ_c is a regulation parameter (in our implementation we use a value of 0.01), and $H_f(u)$ and $H_b(u)$ are the probabilities of the current foreground and background regions including pixels with feature values in bin u, respectively. A higher value of the relearning weight c_u (which ranges between 0 and 1) indicates that it is safe to use the current foreground to update the reference model for this

*Corresponding author: Jingjing Xiao, Email: shine636363@sina.com

histogram bin, since this feature value is highly likely to represent the target rather than the background.

C. *Updating the target reference model*

We can now use the histogram models of the current foreground region, together with the relearning weights, to stably and robustly allow updating of the target reference model. For each bin of each model we update as:

$$\hat{H}_{ref}^{colour}(\zeta) = (1 - c_u^{colour})H_{ref}^{colour}(\zeta) + c_u^{colour}H_f^{colour}(\zeta) \quad (27)$$

$$\hat{H}_{ref}^{hog}(\zeta) = \left(1 - c_u^{hog}\right)H_{ref}^{hog}(\zeta) + c_u^{hog}H_f^{hog}(\zeta) \quad (28)$$

where h_{ref} represents the reference appearance model. In other words, those histogram bins which are most dissimilar to the current background distribution make the biggest contribution to the target model-updating. After the target models have been updated, a normalization stage is also carried out to ensure that probabilities add up to unity over the resulting updated reference histograms. The overall target model re-learning scheme is summarized in Fig.5.

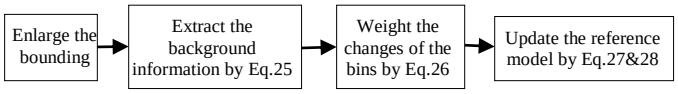

Figure 5. Target reference model relearning scheme

VI. Handling Occlusions

When tracking targets that move past cluttered backgrounds, we also incorporate an additional robustifying measure, which detects when the target is being temporarily occluded, and modifies various steps of the tracking algorithm accordingly to prevent instability.

When the target is occluded, it is desirable to maintain as broad a spread of particles as possible, since this helps to effectively search for and redetect the target, by spanning the possible locations where the target might re-emerge. Therefore, when an occlusion situation is detected, we switch off the two-step drift prevention mechanism described in section IV. Additionally, when the target is likely being occluded, there is an increased danger of erroneously relearning non-target pixels (either background pixels or occluding object pixels) into the target reference model. Therefore, when occlusion situations are detected, we also switch off the online target relearning procedure described in section V. Both the drift prevention and the target relearning procedures are switched back on again, once the occlusion situation is judged to have ended, i.e. after the target is judged to have re-emerged from behind an occluding object.

In our algorithm, occlusion is detected using the simple procedure proposed by [20]. First, the overall target location is estimated according to equation 1. Next, a bounding box is positioned at this location and a new histogram is formed for the feature values of bounding box pixels. This is then compared against the target reference model, using equations 17, yielding an overall likelihood for the tracker. A state of occlusion is assumed when the overall likelihood falls below a predetermined threshold. The overview of the whole proposed method is shown in Tab. 1.

Table. 1 Overview of the proposed tracker

Initialize the tracker with one bounding box X_o. Perform the following steps:
1. **Propagate** particles around the target from the last frame by Eq.3.
2. **Observe** the weight of each particle, considering both colour and shape features, Sec.III.
3. If **unoccluded** (overall likelihood is above a threshold),
i) Two-step estimation. Output target position, Sec.IV.
ii) Update the model according to the contextual information Sec.V
Else
Estimate the target position as proposed in [20].

VII. Experiments and Results

A. *Performance evaluation methodology*

We have evaluated the performance of the proposed method using the publicly available VOT benchmark dataset [19], according to the performance evaluation methodology established in [24]. The dataset comprises 11 videos, in which a variety of different target objects must be tracked under a variety of challenging conditions. For comparative evaluation, we have tested the same dataset on six other well respected trackers in [10]. According to the findings of [24], performance of tracking algorithms is well characterized by two key parameters: accuracy and robustness. Accuracy A_k is defined in terms of the degree of overlap between the ground truth bounding box region GT_k, and the estimated bounding box region output by the tracking algorithm, known as "tracker truth", TT_k:

$$A_k = \frac{GT_k \cap TT_k}{GT_k \cup TT_k} \quad (29)$$

Robustness is defined in terms of the number of times that the tracker loses the target, with respect to a specified minimum accuracy threshold:

$$R_\tau = \|\{k|A_k > \tau\}_{k=1}^{N}\|/N \quad (30)$$

where τ denotes the threshold of the accuracy for A_k, and N is the total number of total frames.

The remainder of this results section proceeds as follows. Subsections *B* and *C*, illustrate the key functionalities (adaptive feature weighting and online target relearning) of our proposed algorithm, by analyzing its behavior during two example sequences which exhibit different kinds of scene "attributes" (tracking difficulties). Subsection *D* shows the results of comparing the performance of our proposed method against the seven other comparison methods, over all nine example videos.

B. *"Gymnastics" sequence*

In this sequence the target object is a tumbling gymnast (see Fig. 6) which undergoes very large self-deformation with rapid and erratic motion, and is tracked against a severely cluttered background.

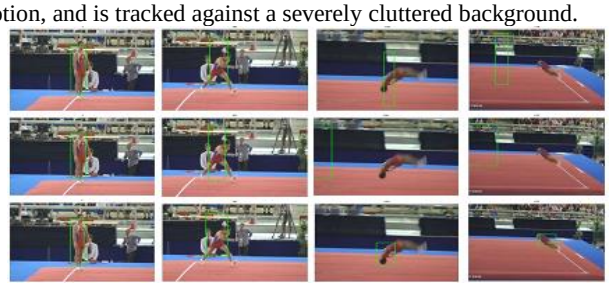

Figure 6. Frames 1, 90, 150, 180 (left to right) of the "Gymnastics" sequence, featuring a rapidly changing target object. The green bounding boxes show the results of particle filter tracking with: no target model relearning (top row); target model relearning using method of [17] (middle row); our proposed method for target model relearning (bottom row).

Fig.6 illustrates how our proposed online target model relearning method enables the tracker to cope with a target object which rapidly changes its appearance. The target exhibits very extreme shape deformations, in addition to less obvious (but still significant) changes in size, colour and illumination. Clearly, methods which do not perform online target model relearning (top row) are likely to fail under such conditions. However, it is interesting to note that target update methods relying only on foreground pixels (middle row [17]) can fail even earlier than no target relearning at all. The bottom row of Fig. 6 shows how our method can successfully track the rapidly changing target, by using a target model relearning scheme which considers pixel statistics from both the current target and also the current local background image regions.

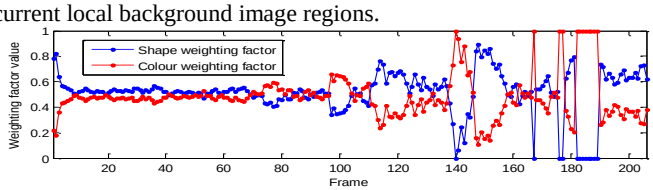

Figure.7 Variation of feature weights during Gymnastics sequence

*Corresponding author: Jingjing Xiao, Email：shine636363@sina.com

Fig. 7 illustrates the adaptive feature weighting scheme which we use for data fusion of multiple features. The figure plots the values of the weights for the colour (red) and shape (blue) features, as they vary frame by frame over the course of the image sequence. The gymnast begins with a short run (frames 0-4, during which the shape feature dominates, probably due to background clutter), and then stands still for a period (frames 5-95) during which the algorithm exhibits no significant preference between the features, reflected by weighting factors close to 0.5 for both features. In frames 95-110, the gymnast again begins running. This rapid motion results in the algorithm devaluing the shape feature and weighting in favour of the colour feature. In contrast, in frames 110-140, the gymnast is still running, but the cluttered background scene shares similar colours with the gymnast's uniform, while the overall shape of the athlete does not vary very much. This causes the algorithm to devalue the colour feature and weight more in favour of the shape feature. In frames 141-190, the gymnast is performing tumbling through the air. He occasionally passes through individual frames for which the shape feature becomes very weak (due to rapid shape and orientation changes), resulting in sharp spikes in the colour weighting.

Fig. 8 shows robustness versus accuracy-threshold curves (described in subsection A) for several variants of the particle filter tracker. Using the Gymnastics sequence, we have compared our proposed adaptive two-feature tracker (red) against: colour only (blue); shape only (green); colour and shape with equal, non-adaptive weighting factor μ_d = 0.5 (black).

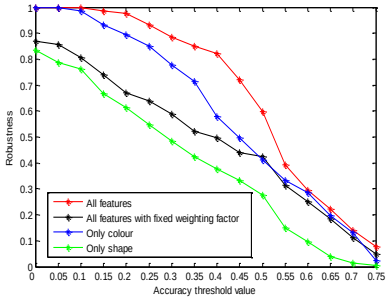

Figure 8. Robustness versus accuracy threshold curves of four tracker variants for Gymnastics sequence.

Fig.8 suggests that the colour feature outperforms the shape feature for the Gymnastics sequence. This is probably due to the extremely large and rapid shape changes exhibited by the target object throughout the sequence. It is interesting to note that a naive (equally weighted and non-adaptive) feature fusion method actually delivers significantly worse results than colour tracking alone. This supports our assertions in sections I.*B* and III.*C* that data fusion must be continuously adaptive, and weight in favour of the most discriminative feature in each frame. Naive (equally weighted) feature fusion (e.g. either either additive or multiplicative) will often fail, because false negative particle weights from a poorly performing feature will damage true positive particle weights from a good feature.

C. "David" sequence

In the David sequence Fig. 9, the tracked target object is a human face, which moves under conditions of very severe illumination changes, which are likely to challenge any tracking methods based on colour features.

Figure 9. Frames 1, 50, 100, 150 (left to right) of the "David" sequence, which requires face tracking under conditions of severe illumination change. The green bounding boxes show the results of particle filter tracking with: no target model relearning (top row); target model relearning using method of [17] (middle row); our proposed method for target model relearning (bottom row).

Clearly, methods which make use of colour intensities, and which do not adaptively relearn the target model online (top row in Fig. 9) will not be able to continue tracking under such severe illumination change. Additionally, recent and well-known methods for target relearning which are based on target region pixels alone, also fail (middle row in Fig. 9). Our proposed target relearning method, which compares feature values in both the target and background image regions, successfully tracks throughout the image sequence (bottom row, Fig. 9).

The illumination changes clearly cause some difficulties for the colour feature target model. However, unlike the Gymnastics sequence (where the camouflaging devalues the discriminating power of the colour feature modality), in the David sequence the colour of the face remains quite distinct from the background in most frames. Therefore, it is not obvious which feature is most discriminatory. This is reflected in Fig.10, which plots the weighting factors for both colour (red) and shape (blue) features. Throughout the sequence, both the colour and shape features share similar weighting factor ranges and do not often deviate far from 0.5 in most frames.

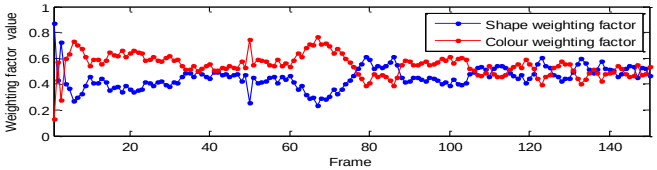

Figure 10. Variation of feature weights during David sequence

Fig. 11 shows the robustness versus accuracy-threshold curves (described in subsection A) for several variants of the particle filter tracker. Similarly to the Gymnastics sequence, our adaptive feature weighting method outperforms the single feature trackers as well as the naïve static equally-weighted feature fusion method. Due to the severe illumination changes, the shape feature is always more discriminatory than the colour feature. In support of our previous assertions, the naïve (equally weighted) multiple-feature method performs no better than the shape feature alone.

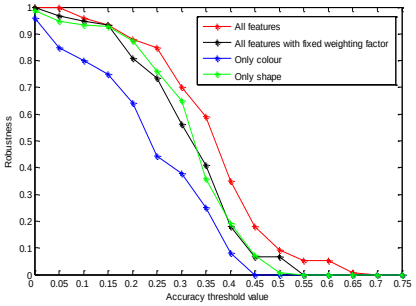

Figure 11. Robustness versus accuracy threshold curves of four tracker variants for David image sequence.

D. Comparison between the proposed method and other state-of-the-art trackers from the literature

We have tested another nine videos from the benchmark dataset [19] on six other well respected trackers from the literature, selected from strongly performing methods in [10]. For comparison, we selected the particle filter method PF [20], from which all histogram-based particle filter methods, including our own, are derived. We also select Struck [36], which ranked first place in [10]. We also select LGT [23] which has recently emerged as one of the most robust trackers published anywhere in the literature. We also select L1 [11], CSK [12], and IVT [3], which all reported excellent performance in [10], summerized in Tab.2.

*Corresponding author: Jingjing Xiao, Email：shine636363@sina.com

Table 2 Summarization of compared tracking algorithms.

Name	Feature	Model adaptation
PF [20]	Colour histogram	Gradient descent
LGT [23]	Intensity histogram, Optical flow, convex envelope	Cross constraint in coupled-layer model
Struck [36]	Haar	Template replacement
L1 [11]	Sparse representation	Template replacement
CSK [12]	Intensity hisrogram	Gradient descent
IVT [3]	Covariance matrix	Incremental update

Each sequence from the benchmark dataset is associated with one or more "attributes" (types of tracking difficulty), as defined in Tab. 3.

Table 3 List of the attributes annotated to test sequences.

Attr.	Description	Attr.	Description
IV	Illumination Variation	**OCC**	Occlusion
SV	Scale Variation	**MB**	Motion Blur
DEF	Deformation	**FM**	Fast Motion
IPR	In-Plane Rotation	**BC**	Background Clutter
OPR	Out-of-Plane Rotation	**LR**	Low Resolution

Tab. 4 shows the "tracking centre error" for each algorithm, averaged over all frames. At each frame, the tracking centre error is defined as the distance (in units of pixels) between the ground truth target centre and the centroid of the target bounding box output by the tracking algorithm. In this table, smaller values indicate superior performance. Tab. 5 shows the accuracy (according to Eq.22) for each algorithm, averaged over all frames of all video sequences. In this table, larger values indicate superior performance.

Table. 4 Comparison results of tracking center errors

Name	Attributes	Ours	PF [20]	LGT [23]	Struck [36]	L1 [11]	CSK [12]	IVT [3]
Bolt	OCC,DEF IPR,OPR	13	42	11	349	384	401	378
Cup	BC	5	14	5	24	3	62	3
Face	OCC	16	17	14	26	8	5	23
Bike	OCC,BC	9	38	52	6	52	62	61
Subway	OCC, DEF, BC	9	145	6	8	150	164	136
Car Scale	SV, OCC, FM, IPR, OPR	24	16	54	33	93	83	15
Walking	SV, OCC, DEF	4	79	6	8	2	7	3
Jogging	OCC, DEF, OPR	12	13	92	73	106	135	89
Crossing	SV, DEF, FM, OPR, BC	10	41	6	121	58	9	4
Mean error over all sequences		11	45	27	72	95	103	79

Table. 5 Comparison results of tracking accuracy

Name	Attributes	Ours	PF [20]	LGT [23]	Struck [36]	L1 [11]	CSK [12]	IVT [3]
Bolt	OCC,DEF IPR,OPR	0.56	0.24	0.42	0.02	0.02	0.02	0.01
Cup	BC	0.70	0.55	0.60	0.56	0.79	0.37	0.79
Face	OCC	0.63	0.51	0.60	0.61	0.77	0.87	0.53
Bike	OCC,BC	0.44	0.28	0.31	0.50	0.43	0.25	0.44
Subway	OCC, DEF, BC	0.52	0.09	0.53	0.67	0.16	0.19	0.12
Car Scale	SV, OCC, FM IPR, OPR	0.47	0.35	0.43	0.42	0.56	0.41	0.64
Walking	SV, OCC, DEF	0.69	0.30	0.48	0.45	0.66	0.54	0.78
Jogging	OCC, DEF, OPR	0.57	0.55	0.09	0.23	0.16	0.18	0.16
Crossing	SV, DEF, FM OPR, BC	0.52	0.31	0.55	0.20	0.18	0.48	0.31
Mean accuracy over all sequences		0.57	0.35	0.45	0.41	0.41	0.37	0.42

In both tables, the best performance for each row is shown in red, and the second best performance is shown in green. According to both metrics, our proposed method significantly outperforms all the other methods when performance is averaged over the entire data set, while the (much more complicated) LGT method [23] also outperforms most other methods to take second place overall. The attributes associated with each benchmark video suggest that our proposed method is particularly robust against occlusions, target deformations, and out of plane rotations.

Note that our proposed method does not perform best for many of the individual video sequences. For the tracking error metric, the proposed method wins first or second place in 4 out of 9 test videos. For the tracking accuracy metric, the proposed method wins first or second place in 5 out of 9 test videos. Some of the comparison methods perform extremely well in a few videos, but also perform extremely badly in other test videos. This suggests that such algorithms are, in a sense, overfitted to certain kinds of tracking situations, but underfitted to others. In contrast, the strength of the proposed tracker appears to be its consistently strong performance across many different kinds of tracking situation.

Fig. 12 shows robustness versus centre-error threshold curves for each tracker. Fig. 13 shows robustness versus accuracy curves for each tracker. According to both performance metrics, our proposed method clearly and significantly outperforms all of the other methods. Fig. 14 illustrates performance of each tracker on selected frames from each benchmark video. To handle the variations of colour or shape feature, the algorithm adaptively fuses different features and updates them during the tracking, which is demonstrated to achieve good performance in *Face*, *Bike*, *Car scale* and *Jogging*. In the cluttered scenes, i.e. *Bolt*, *Cup*, *Subway* and *Walking*, the proposed method benefits from two-step estimation to prevent excessive particle drifting to the camouflages, while other trackers fail in the local optimal regions.

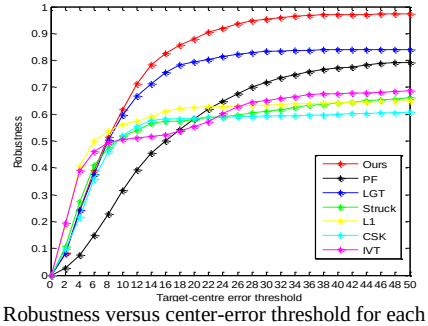

Figure 12. Robustness versus center-error threshold for each tracker.

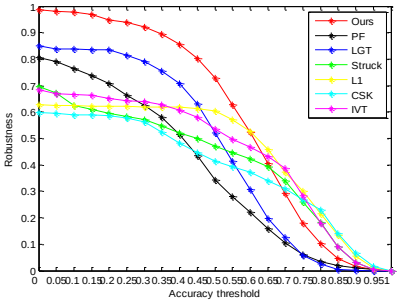

Figure 13. Robustness versus accuracy threshold for each tracker.

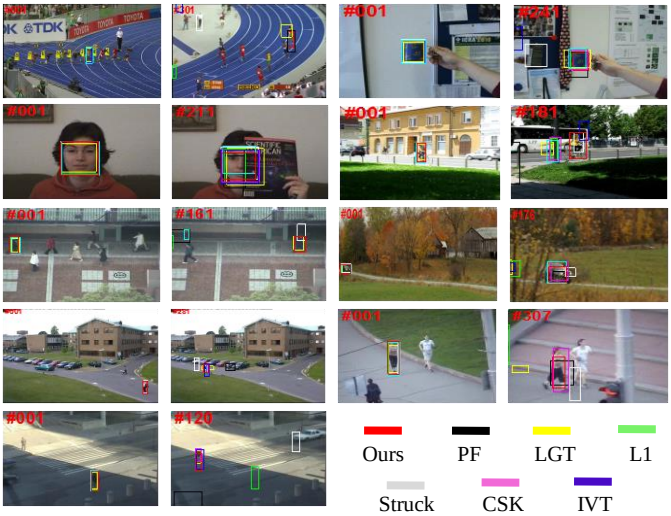

Figure 14. Outputs of all trackers on selected frames of each sequence.

*Corresponding author: Jingjing Xiao, Email：shine636363@sina.com

V III. Conclusion

This paper has revisited the comparatively simple histogram-based particle filter approach of [18] and [29], and demonstrated how it can be enhanced to achieve competitive performance against the most robust of complex modern methods. The proposed enhancements include: i) a continuously adaptive data fusion method for optimally combining multiple features; ii) A more robust method for continuous re-learning of targets which change their appearance, while avoiding the accidental relearning of background features into the target models; iii) A two-step estimation method to prevent excessive particle drifting. We have tested the proposed enhanced, multi-feature particle filter tracker against a number of state-of-the-art tracking methods. Experiments suggest that the proposed method can outperform the leading methods from the literature on such data. We have argued and presented supporting evidence that effective fusion of multiple features or modalities of visual data requires a continuously adaptive process, which can weight in favour of whichever modality is most discriminating in the current frame. We have also argued that such feature weighting mechanisms must take account of image pixel data from the local background region as well as the currently estimated target region. We have also argued that similar reasoning is necessary to enable robust methods of online target model relearning, which can avoid instabilities due to erroneous learning of background pixels into the target model.

Acknowledgment

This work was supported in part by EU H2020 RoMaNS, 645582, and EPSRC EP/M026477/1. NSF of PR China, 61171136.

References

[1] A. Bhattacharyya, "On a Measure of Divergence between Two Multinomial Populations," *Sankhya*, 1946.

[2] D. Comaniciu, M. Peter, and R.Visvanathan. "Kernel-based object tracking," *In PAMI*, 2003.

[3] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. "Incremental Learning for Robust Visual Tracking." *In IJCV*, 2008.

[4] D. Comaniciu, R. Visvanathan, and M. Peter, "Real-time tracking of non-rigid objects using mean shift." *In CVPR*, 2000.

[5] D. J. Duff et al. "Physical simulation for monocular 3D model based tracking." *In ICRA*, 2011.

[6] E. Maggio, S. Fabrizio, and C. Andrea. "Adaptive multifeature tracking in a particle filtering framework." *IEEE TCSVT*, 2007.

[7] F. Ennesser, and G. Medioni. "Finding Waldo, or focus of attention using local colour information." *In PAMI.* 1995.

[8] G. Hinton, "Products of experts," *ICANN*, 1999.

[9] G. R. Bradski, Computer vision face tracking for use in a perceptual user interface. 1998.

[10] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark". *In CVPR*, 2013.

[11] X. Mei and H. Ling. "Robust visual tracking using l1 minimization." *In CVPR*, 2009.

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the Circulant Structure of Tracking-by-detection with Kernels," *In ECCV*, 2012.

[13] J. Xiao, R. Stolkin, and A. Leonardis. An enhanced adaptive coupled-layer LGTracker++. *ICCV workshop VOT*, 2013.

[14] J. Chang, and J. M. Fisher. "Topology-Constrained Layered Tracking with Latent Flow". *In ICCV*, 2013.

[15] J. Li and C.-S. Chua, "Transductive local exploration particle filter for object tracking," *Image and Vision Computing*,2007.

[16] J. Vermaak, D. Arnaud, and P. Patrick. "Maintaining multimodality through mixture tracking." *In ICCV*, 2003.

[17] J. Martınez-del Rincon, C. Orrite, and C. Medrano, "Rao–blackwellised particle filter for colour-based tracking," *Pattern Recognition Letters*, 2011.

[18] K. Nummiaro, K.M. Esther and V. G. Luc, "A color-based particle filter." *First International Workshop on Generative-Model-Based Vision*. 2002.

[19] The VOT 2013 evaluation kit: http://www.votchallenge.net.

[20] K. Nummiaro, K.M. Esther and V. G. Luc, "An adaptive colour-based particle filter," *Image and Vision Computing*, 2003.

[21] Kristan et al. "The Visual Object Tracking VOT2013 challenge results." *ICCV VOT Challenge workshop, 2013.*

[22] L. S. Hanxuan Yang, Feng Zheng, Liang Wang, and Zhan Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, 2011.

[23] L. Čehovin, M. Kristan and A. Leonardis, "Robust Visual Tracking Using an Adaptive Coupled-Layer Visual Model," *In PAMI* , 2013.

[24] L. Čehovin, M. Kristan, and A. Leonardis, "Is my new tracker really better than yours", *In WACV*. 2014.

[25] M. Talha, and R. Stolkin. "Particle filter tracking of camouflaged targets by adaptive fusion of thermal and visible spectra camera data*", IEEE Sensors Journal*, 2013.

[26] M. Isard, and B. Andrew. "Condensation—conditional density propagation for visual tracking." *IJCV*. pp: 5-28, 1998.

[27] N. Dalal, B. Triggs. "Histograms of oriented gradients for human detection". *In CVPR*, 2005.

[28] O. L. Junior, D. Delgado, V. Gonçalves and U. Nunes, "Trainable classifier-fusion schemes: an application to pedestrian detection". *In Intelligent Transportation Systems.* 2009.

[29] P. Pérez, et al. "Color-based probabilistic tracking." *In ECCV*,. 2002.

[30] P. Perez, V. Jaco and A. Blake. "Data fusion for visual tracking with particles." *Proceedings of the IEEE*, 2004.

[31] P. Brasnett, et al. "Sequential Monte Carlo tracking by fusing multiple cues in video sequences." *Image and Vision Computing*, 2007.

[32] J. Xiao, R. Stolki, A. Leonardis, "Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models", *In CVPR*, 2015.

[33] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features*". In PAMI*, 2005.

[34] R. Stolkin, I. Florescu, and G. Kamberov. "An adaptive background model for camshift tracking with a moving camera." *Proc. International Conference on Advances in Pattern Recognition*, 2007.

[35] S.-K. Weng, C.-M. Kuo, and S.-K. Tu, "Video object tracking using adaptive kalman filter," *Journal of Visual Communication and Image Representation*, 2006.

[36] S. Hare, A. Saffari, and P. H. S. Torr. "Struck: Structured Output Tracking with Kernels." *In ICCV*, 2011.

[37] M. Kristan, S. Kovačič, A. Leonardis, et al. "A two-stage dynamic model for visual tracking," *IEEE Systems, Man, and Cybernetics, Part B*, 2010.

[38] J. Xiao, M. Oussalah, "Robust model adaptation for tracking with online weighted color and shape feature", *4th IEEE International Conference on Image Processing Theory, Tools and Applications*, France, 2014.

*Corresponding author: Jingjing Xiao, Email：shine636363@sina.com