

On the use of contextual information for robust colour-based particle filter tracking

Jingjing Xiao¹, Mourad Oussalah²

1: University of Birmingham, School of Engineering, EECE, Birmingham, UK

2: University of Oulu, Centre for Ubiquitous Computing, Oulu, Finland

Abstract— Color-based particle filters have emerged as an appealing method for targets tracking. As the target may undergo rapid and significant appearance changes, the template (i.e. scale of the target, color distribution histogram) also needs to be updated. Traditional updates without learning contextual information may imply a high risk of distorting the model and losing the target. In this paper, a new algorithm utilizing the environmental information to update both the scale of the tracker and the reference appearance model for the purpose of object tracking in video sequences has been put forward. The proposal makes use of the well-established color-based particle filter tracking while differentiating the foreground and background particles according to their matching score. A roaming phenomenon that yields the estimation to shrink and diverge is investigated. The proposed solution is tested using publicly available benchmark datasets where a comparison with six state-of-the-art trackers has been carried out. The results demonstrate the feasibility of the proposal and lie down foundations for further research of complex tracking problems.

Index Terms—Object tracking, Video Analysis, Scale modification, Background learning.

I INTRODUCTION

With widespread of multimedia standards like MPEG-4, large amount of surveillance cameras have been deployed in cities, public buildings, motorways, etc. This resulted in substantial increase of multimedia data, which, in turn, rendered the task of robust automatic tracking system from video sequences of paramount importance because of the cost of manual check. Object tracking, as a fundamental problem in computer vision, can be further used in video compression, video retrieval, interactive video, scene composition, etc. [1]. This motivates extensive research in recent years to improve the efficiency of automatic target tracking and event recognition from video sequences [2]. However, visual tracking still poses many open challenges, e.g., background clutter, occlusion, fast movement, variation of illumination, object scale change and deformation [3-4]. To build effective mathematical models, numerous cues have been explored to represent target, including, motion, geometry, shape and colour [5]. Models based on a combination of features have also been explored in [6-7]. In a standard tracking model, the basis is to use a reference model, which can be any image patch or manual inputs of contour, colour histogram of object of interest, among others, that describes the appearance of the underlying object. Several approaches have been put forward that utilize this colour feature in video tracking (i.e. CamShift

[8] MeanShift [9], Kalman Filter [10], Particle Filter [11] etc.). As Particle Filters have distinguished properties in dealing with multi-modal visual tracking problem of general non-linear and non-Gaussian systems without any assumption about the dynamics and shape of the conditional density, it has been used extensively in recent past [12-13]. In this course, the key is to calculate the similarity between the target colour histogram at the region around the target estimate (referred to tracker scale) and that of the ground truth (reference mode) [14-15]. Therefore, the (estimated) region (or tracker scale) around the target location estimate plays key role in the estimation process. If the size of such region becomes sufficiently small, the target may be lost by the tracking system, as it may yield no matching between the target and ground truth histograms. Similarly a wide (estimated) region may render the influence of the background pixels more dominant causing the tracker to become more easily distracted by background clutter, which, in turn, may lead to target loss as well. In addition, the appearance of the target might change according to self-deformation and illumination variations, which also renders the similarity to the reference model biased. To handle these newly emergent/disappeared features, the appearance model also needs to be updated, accordingly.

In this paper, likewise works in [14-21], we deliberately confine our study to a single object tracking, although we acknowledge the extensibility of the approach to multiple objects after handling the possible data association problem as in [22-23]. The main contributions of this paper are twofold. First the original color based particle filter tracker has been extended by accounting for background information around the estimated region of the target. Second, roaming density criteria were put forward by eliciting marginal density function, to model particles' shrinking behavior, and then, if necessarily, refine particles' weights and, accordingly, the estimation is updated. Especially, contextual information is accounted for to enhance the robustness of the tracker. The algorithm is tested using a set of benchmarked dataset to ease comparison with other state of art trackers.

II RELATED WORK

Recently, the adaption of tracker scale and appearance model has been studied, extensively. The work in [5], [11], [14] considers the scale of the tracker as part of the state vector to be estimated by the particle filter at each sample. Although

such approach sounds appealing, the influence of the appearance model with respect to target cannot be neglected. Indeed, since the particle filter measurement relies on matching the appearance model (colour distribution) around the estimated position to that of the target, it is easy to see that scenarios of tiny variations of appearance models make the scale parameter estimation biased and yields possible local optimality because both small and large tracker scale would lead to almost the same result in terms of similarity between appearance model around the estimate and that of the appearance of target (reference) model.

Several approaches have been put forward for the purpose of appearance model update, e.g., linear combination of previous and current estimations of template model at the end of each filter iteration [11], decision rule-based method [15], Rao-Blackwellised particle filter (RBPF) like approach [5], among others. However, all these updating stages are solely based on the information extracted from the estimation where any inaccurate estimation is likely to result in serious drifting problem. Other researchers attempted to extract contextual information as an aid for tracking, e.g., Wang et al.'s on-line updating of the appearance model by selecting the discriminative features with the aid of existing background particles [24]. Stolkin et al. [25] attempted to extract the information from the background and automatically adjust the relative importance of the two subsequent features (i.e., thermal and colour features), while background information was used to re-weight the existing features.

III COLOR BASED PARTICLE FILTER

The key idea of particle filter is to represent the posterior density function by a set of random samples $(s^{(i)}, \omega^{(i)}, i = 1, N)$, where $s^{(i)}$ represents some hypothetical state of the object of interest, referred to i^{th} particle, with associated weight $\omega^{(i)}$ such that $\sum_{i=1}^N \omega^{(i)} = 1$. While the overall estimation of the state X of object of interest from its particles is performed using the expectation operator over the set of particles; namely, $X = \sum_{i=1}^N s^{(i)} \omega^{(i)}$.

In our algorithm, the target is modelled by a rectangular region, which defines the tracker scale:

$$s = \{P^x, P^y, H^x, H^y\} \quad (1)$$

where P^x, P^y represent the x-y coordinates of the center position of the tracker, H^x, H^y stand for the region width and height, respectively, as shown in the Fig. 1.

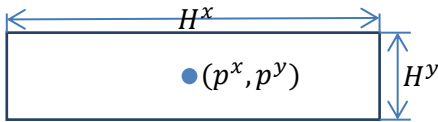


Fig. 1 Tracker bounding box

It is commonly assumed that the state vector s is latent and/or unobservable, estimated from a set of (noisy) observations Z . We denote by Z_k the observation at current time k , and $Z_{1:k}$ the cumulative set of measurements up to and including time k .

The estimation of the posterior in particle filter uses three main steps: prediction, update and re-sampling. First, prior information is utilized to propagate and predict the potential distribution of samples. The update operation uses the measurement to modify the predicted probability density function (pdf). By using the principle of importance sampling, the weights are chosen and allocated to the corresponding samples [26]. Then, the update state is computed by averaging over the set of particles. More specifically, similarly to traditional color based tracking [11], the dynamic of the state model is described by a constant motion model:

$$s_k = A s_{k-1} + v_{k-1} \quad (2)$$

where $v_k \sim G(0, R)$ is a zero-mean Gaussian noise with a constant variance-covariance matrix R , and A is an identity matrix of size four.

In the update stage, the measurement Z_k is used to modify the prior density by evaluating the similarity between the ground truth and the prediction. In other words, instead of taking the entire image as a measurement, one only restricts to the image region specified by the state vector $s_k^{(i)}$ (region centred on (P_k^{ix}, P_k^{iy}) and whose width and height are H_k^{ix} and H_k^{iy} , respectively) for the i^{th} particle. Next, the associated probability distribution $h_{s_k^{(i)}}$ is constructed using the m -color histogram of the above region. On the other hand, the probability model corresponding to the reference model is also computed. This may refer to the first frame or the image request containing the object to be tracked, so that the rectangular region that fully delimits the object of interest is fully quantified using its m -bins histogram based probability distribution h_{ref} . Next, the similarity $D(h_{s_k^{(i)}}, h_{ref})$ between the reference target model h_{ref} and the candidate target model $h_{s_k^{(i)}}$ according to state estimation $s_k^{(i)}$ is calculated using commonly employed Bhattacharyya distance [27]:

$$D^2(h_{s_k^{(i)}}, h_{ref}) = 1 - \sum_{j=1}^{N_{bin}} \sqrt{h_{s_k^{(i)}}(j) \cdot h_{ref}(j)} \quad (3)$$

where $h_{s_k^{(i)}}(j), h_{ref}(j)$ indicate the value of bin j in the histogram of i^{th} particle and reference model, respectively, normalized in the unit interval. N_{bin} is the total number of the bins. Finally, particles' weights are extracted from Bhattacharyya distance as:

$$\omega^{(i)} = e^{-\lambda_* D^2(h_{s_k^{(i)}}, h_{ref})} \quad (4)$$

where λ_* is some positive constant parameter. Weights in (4) are then normalized by a simple division over $\sum_{i=1}^N \omega^{(i)}$, say,

$$\omega'^{(i)} = \omega^{(i)} / \sum_{i=1}^N \omega^{(i)} \quad (5)$$

Next, the estimated state is obtained by averaging over the set of all particles:

$$\hat{s}_k \approx \sum_{i=1}^N \omega'^{(i)} s_k^{(i)} \quad (6)$$

The choice of parameter λ_* governs the number of particles with high weights that can be generated. For instance, choosing high value of λ_* would substantially increase the number of particles whose weights are close to zero value, while choosing λ_* very small would yield marginally equal

weights. Setting $\lambda_*=1$ sounds a cautious attitude that avoids both overfitting and under-fitting phenomena. Next a resampling stage is carried out to discard small weight particles in the same spirit as in [13].

IV MODEL ADAPTATION

We first differentiate foreground and background particles. The algorithm uses the distribution of those foreground particles to model particles' roaming behavior for tracker scale adaption while the contextual information extracted from the surrounding is utilized for appearance model adaptation, as described in the following subsections.

A. Scale adaption

In order to ensure efficiency in object tracking performance, the (estimated) scale of the tracker should be resized at each frame in order to accommodate the dynamic and inherent variations of shape of tracking object (e.g., rotation, moving forward or backward from camera view). Especially, when the tracker scale is too small, the kernel can *roam* around on the object, and leads to a poor object localization, which can cause tracking failure.

Especially, provided that the genuine target is not of uniform color, the roaming phenomenon occurs because of an erroneous estimation that either shift the bounding box estimates to the background region or the bounding box regions become marginally small. This raises the importance of reliable identification of foreground regions as well as monitoring of the target scale estimates, issues that will be detailed in the next section.

1) Roaming detection

The key idea to tackle the roaming phenomenon is to avoid its occurrence. For this purpose, the rationale is to account for contextual information in order to favor higher-weight particles. Three rational criteria have been pursued: distinguishing foreground from background region, Accounting for only high-weight particles and Taking into account the distribution of the particles in the foreground region. In order to differentiate the foreground and background particles, the matching scores $\{\omega_k^{(i)}\}$ is employed. Especially, a particle is regarded as foreground if its associated weight value $\{\omega_k^{(i)}\}$ is beyond some predefined threshold value ω^* . The latter is chosen such that

$$\omega_k^* = \mu_s * \max_i \{\omega_k^{(i)}\}, \quad (7)$$

where $\{\omega_k^{(i)}\}$ are computed according to Eq. 4-5 and μ_s is a fixed weighting factor ranging in the unit interval. Typically, $\mu_s = 1$ would restrict the number of particles to only the one with the highest weight, while $\mu_s = 0$ makes all particles to pass the threshold test, therefore a balance should be provided by avoiding low number of particles, which would make the non-linear approximation of particle filter very weak and the high number of particles which would worsen the roaming behavior. Interestingly, one shall notice that

since λ_* also influences the weight of particle, the dependency between λ_* and μ_s is established. Therefore fixing the value of one parameter to a reasonable default and studying the variation of the other sounds a rational attitude. Discussion regarding the choice of μ_s is reported to experimental section.

Second, the distribution of the underlying foreground particles in terms of spatial representation is investigated. Intuitively, if the covering area of such particles is too small, then it likely induces a shrinking problem, alternatively if it is too large, this would also suggest that the underlying region will likely include some background pixels as well, which, in turn, may trigger the roaming phenomenon. Therefore, we define the concept of roaming threshold value as

$$D_r = \frac{\sqrt{\sum_{i=1}^N \|\omega_k^{(i)} - \bar{\omega}_k\|^2 \delta(\omega_k^{(i)} - \omega_k^*)}}{S_{roam}}, \quad (8)$$

where S_{roam} corresponds to the (geometrical) area of the foreground particles, $\bar{\omega}_k$ is the mean weight of these particles and N is the total number of particles. More formally, let L be the set of particles whose weights is beyond ω_k^* :

$$L = \{S_{\psi(1)}, S_{\psi(2)}, \dots, S_{\psi(M)}\}, \quad (9)$$

where M is the total number of such particles and ψ is some permutation of indices $\{1, 2, \dots, N\}$ such that all particles in set L meet the threshold criterion. Let R^* be the rectangular region that includes all the sub-regions R_{S_k} associated to the foreground particles whose weights are beyond threshold ω_k^* . In the same spirit as (1), let

$$S_k^{(\psi(i))} = \{P_k^{x_{\psi(i)}}, P_k^{y_{\psi(i)}}, H_k^{x_{\psi(i)}}, H_k^{y_{\psi(i)}}\}, i = 1, M, \quad (10)$$

be the set of particles that are found to belong to foreground, then

$$S_{roam} = \min_{i=1, M} (P_k^{x_{\psi(i)}} - H_k^{x_{\psi(i)}}) \max_{i=1, M} (P_k^{y_{\psi(i)}} - H_k^{y_{\psi(i)}}) \quad (11)$$

$$\delta(\omega_k^{(i)} - \omega_k^*) = \begin{cases} 1 & \text{if } \omega_k^{(i)} \geq \omega_k^* \\ 0 & \text{Otherwise} \end{cases} \quad (12)$$

$$\bar{\omega}_k = \sum_{i=1}^N \omega_k^{(i)} \delta(\omega_k^{(i)} - \omega_k^*) \quad (13)$$

From (11), higher the number of foreground particles, likely wider the entity S_{roam} . From (8), a threshold, say, T_r , was used to detect roaming phenomenon; namely, the roaming occurs whenever

$$D_r \leq T_r \quad (14)$$

Given the arguments provided earlier, the threshold T_r is chosen inversely proportional to the number of particles in the foreground:

$$T_r = \frac{f_s}{S_{rom_{initial}} M} \quad (15)$$

Where $S_{rom_{initial}}$ corresponds to the initial value of S_{roam} after the first estimation (first frame) and f_s the (maximum) scaling factor of images in the video sequence. The rationale behind the preceding is that the (target) shrinking and roaming phenomenon rather occurs at later stages, therefore, it is fair to assume the estimation of S_{roam} at the beginning as relatively reliable. The parameter f_s quantifies the variation of the distance object – camera, which in our case was set to unity. If there is no roaming problem, say, (14) is not held, the algorithm will follow the original method of Eq. 5-6 to

estimate the target status, otherwise, the scale modification step described in the next subsection applies.

2) Scale modification

First one shall consider the *cover*-particle; namely, the minimal-area particle that covers all particles in the foreground region (whose geometrical area coincides with S_{roam}). Let us denote by s_k^{R*} this cover-particle.

After generation of the above modified bounding box associated particle, a new resampling scheme is induced accordingly by resetting the scale of all particles according to the estimated state, and then computing the histogram $h_{s_k^{R*}}$ of this new tracker from the modified estimated state (cover-particle) s_k^{R*} and use Bhattacharyya distance as in (3) to compute the similarity between the histogram associated to ground truth and that of the cover-particle as well as weights according to (4).

B. Appearance adaption

Due to discrepancies caused by changing target appearance a static appearance model is not sufficient to form a robust tracker in real scenarios. To overcome the resulting appearance changes, in the same spirit as [11], one updates the appearance model according to the current estimate.

For this purpose, we first set a threshold T_a for the Bhattacharyya distance $D^2(h_{\omega_{s_k^{R*}}}, h_{ref})$ so that the appearance adaption is triggered only if

$$D^2(h_{\omega_{s_k^{R*}}}, h_{ref}) > T_a \quad (16)$$

Inspired by the work of Wang et al. [24] and Stolkin et al. [22, 25], we first extract an appearance model of background. Second, a likelihood based approach has been employed to update the weights attached to reference model.

1) Extracting background information

Since in the roaming detection section, an approach has been developed to distinguish foreground from background particles, the same test (quantified by expression (9)) can therefore be employed to discriminate background particles, say, $s_k^{b_i}$, as well. Indeed, all particles whose weights are below the threshold ω_k^* are assumed background. Next, the appearance model of the background is calculated as the sum of the colour-histogram of its associated particles:

$$h_b = \sum_{i_b=1}^{N_b} h_{s_k^{b_i}} \quad (17)$$

where $h_{s_k^{b_i}}$ is the colour-model of the $s_k^{b_i}$ particle, and N_b stands for the number of background particles.

2) Reference model update

After computing the appearance model of the background using (17), for each bin u , we evaluate the marginal contribution of foreground appearance model with respect to background appearance model, so that such contribution reaches its maximum value one if there is no particles in background or the foreground is highly dominant and takes zero value in the case of absence of foreground particles or strong background dominance. This is quantified using the following:

$$\alpha(u) = \begin{cases} 1 - e^{-\lambda_c \left(\frac{h_f(u)}{h_b(u)} \right)}, & \text{if } h_b(u) \neq 0 \\ 1, & \text{if } h_b(u) = 0 \end{cases} \quad (18)$$

where λ_c is a positive smoothing parameter. The value of $\alpha(u)$ ranges in $[0,1]$. Next, $\alpha(u)$ coefficients are used to adaptively update the appearance of the reference model according to:

$$\hat{h}_{ref}(u) = (1 - \alpha(u))h_{ref}(u) + \alpha(u)h_f(u) \quad (19)$$

where h_{ref} represents the reference appearance model. The latter is normalized by dividing it over the sum across bins. A pseudo-code description of the whole algorithm is given by Table I.

TABLE 1 PSEUDO-CODE OF OVERALL ALGORITHM

Algorithm: Robust model adaption for color-based particle filter tracking with contextual information

Given the sample set $\{s_0^{(i)}\}$ and the target model h_{ref} . Perform the following steps:

1. **Select** N samples from the set S_{k-1} ;
 2. **Predict** each sample from the set $\{s_{k-1}^{(i)}\}$ by linear stochastic differential in (2);
 3. **Observe** the colour distribution:
 - (a) Calculate the colour histogram for each sample of set $\{s_k^{(i)}\}$;
 - (b) Weight each particles $\{\omega_k^{(i)}\}$ using (4);
 4. **Detect** the roaming phenomenon
 - (a) Identify foreground particles
 - (b) Test for roaming occurrence;
 5. If **no shrinking** (roaming test is negative)

Estimate the mean state of the set S_k by Eq. 6 and use systematic resampling;
 - If **shrinking**

Modify the tracker scale size and calculate new weights;
 6. **Update** the appearance model
 - (a) Test whether appearance update is needed
 - (b) If the above test is positive, calculate for each bin u , calculate the appearance mode of the background using (17), and apply the update rule (28-19);
 - (c) Otherwise if the test (16) is negative, the reference model remains unchanged.
-

V EXPERIMENTAL SETUP AND RESULTS

A. Introduction

For evaluation purpose, given the estimated bounding box region ET of the tracker and the ground truth region GT, we used standard precision ($P = (GT \cap ET)/ET$), recall ($R = (GT \cap ET)/GT$) and Overlap ($O = (GT \cap ET)/(GT \cup ET)$).

An initial testing using simulated dataset, which is not covered in this paper provided a roughly good estimate of the threshold value $T_a = 0.1$ as achieving the best performance in terms of precision-recall taking into account the wide set of scenarios of object / background configurations and clutter intensities.

B. Experiment using real videos and comparison with some state of art trackers

To have an overall evaluation, we tested our newly designed algorithm on the explicitly selected sequences from two publicly available benchmark datasets [29] [30], with another 6 state-of-the-art trackers; namely, namely, the original PF tracker [11], adaptive-coupled layer visual model [17], Struck [18], L1-minimization [19], circular structure kernel [20] and Incremental learning tracker [21]. We first test our algorithm on the 16 sequences which have significant scale change, namely, *Bike, Boy, Bolt, Basketball, CarScale, Couple, Crossing, Diving, Tunnel, Skiing, Singer, Fish1, Fish2, Jogging, Gymnastics, Walking*. The result in terms of trade-off curve of success rate versus overlap threshold is shown in Fig. 2.

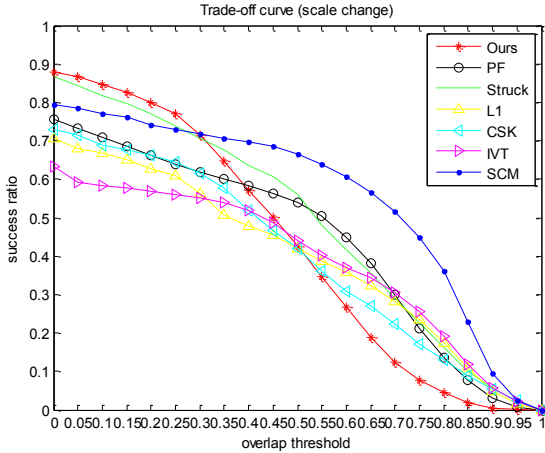


Fig. 2 Trade-off curve of target with scale changes (16 sequences)

Then, we test our algorithm on the 18 sequences with significant appearance change; namely, *Basketball, Bolt, Diving, Tiger1, Tiger2, Trellis, Tunnel, Skiing, Skating, Sunshade, Shaking, Soccer, Fernando, Fish1, Fish2,*

Ironman, Singer, Gymnastics. The result is shown in Fig. 3.

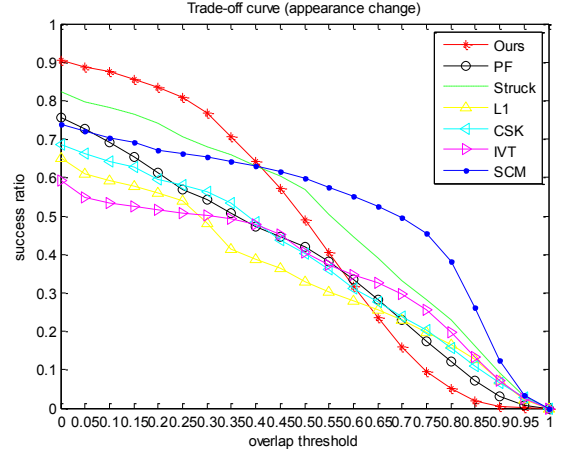


Fig. 3 Trade-off curve of target with scale changes (18 sequences)

Similarly, we have also conducted similar studies on those video sequences that involve *significant shape deformation* (*Basketball, Bolt, Couple, Crossing, David, David3, Dudek, FleetFace, Jogging, Mhyang, Singer2, Skating, Skiing, Subway, Tiger1, Tiger2, Walking, Woman*) with comparison to the six state art trackers in Fig. 4.

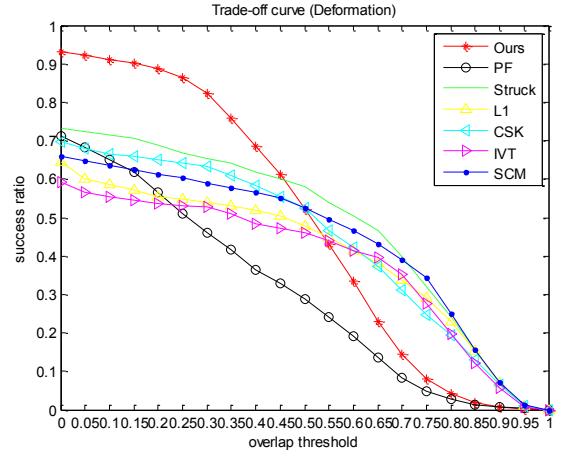


Fig. 4 Trade-off curve of target with shape deformation (18 sequences)

As it can be noticed from the results highlighted in Figs. 2-4, the performance of the developed tracker substantially compete with the state of art trackers and marginally outperform these trackers. Next, average results in terms of centre tracking error across the nine distinct publicly available sequences [30] with comparison to the six state-of-the-art trackers are summarized in Table II. The results show a good performance of our tracker in overall where in 80% of employed dataset, our tracker is either ranked first or second. Especially the best performance is achieved with sequences involving human motion.

TABLE II COMPARISON RESULTS OF TRACKING ACCUARCY

Video Name	Ours	PF [11]	LGT [17]	Struck [18]	L1 [19]	CSK [20]	IVT [21]
Bolt	0.55	0.23	<u>0.42</u>	0.02	0.02	0.02	0.01
Bike	<u>0.44</u>	0.28	0.31	0.50	0.43	0.25	0.43
Basketball	0.53	0.25	<u>0.50</u>	0.09	0.03	0.02	0.33
CarScale	0.53	0.35	0.43	0.42	<u>0.55</u>	0.41	0.63
Crossing	<u>0.53</u>	0.43	0.55	0.31	0.20	0.18	0.48
David3	0.59	<u>0.58</u>	0.25	0.29	0.38	0.50	0.44
Face	0.65	0.51	0.60	0.61	<u>0.77</u>	0.87	0.53
Jogging	<u>0.51</u>	0.55	0.09	0.23	0.16	0.18	0.16
Subway	<u>0.63</u>	0.09	0.54	0.67	0.16	0.19	0.56
Mean acc. over all sequences	0.55	0.36	<u>0.41</u>	0.35	0.30	0.29	0.40

VI CONCLUSION

In this paper, we proposed a robust model adaption method, for both scale and appearance model, which utilizes the contextual information from the background. We use roaming density to model the behavior of foreground particles and modify the kernel according to the distribution of the particles. Simultaneously, the resampling method also changes and adapts to solving the degeneracy problem. In order to avoid appearance model drifting problem, the algorithm re-weights the confidence of the color distribution changes by learning from the present background, which means the color that has a high probability of belonging to the target will be updated relative fast, otherwise it will remain unchanged. We tested our proposed approach on both simulated and publicly available datasets, and demonstrated the significant improvements over several other state-of-the-art trackers. In the future work, we will continue to research on the ways to self-optimization of some parameters involved in this approach and extend the tracker to multiple targets.

ACKNOWLEDGMENT

The first author would like to thank Dr. Rustam Stolkin in School of Mechanical Engineering, University of Birmingham for his discussions with regards to template update stage.

REFERENCES

- [1] K. Hariharakrishnan, and D. Schonfeld, "Fast object tracking using adaptive block matching," *IEEE Transactions on Multimedia*, pp:853-859, 2005.
- [2] M. Szczodrak, P. Dalka, and A. Czyzewski, "Performance evaluation of video object tracking algorithm in autonomous surveillance system," in *Proc. International Conf. Information Technology*. IEEE, 2010, pp. 31-34.
- [3] J. Xiao, R. Stolkin, and A. Leonardis, "An enhanced adaptive coupled-layer LGTracker++," in *ICCVW*, pp. 137-144, 2013.
- [4] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823-3831, 2011.

- [5] J. Martinez-del Rincon, C. Orrite, and C. Medrano, "Rao-blackwellised particle filter for color-based tracking," *Pattern Recognition Letters*, vol. 32, no. 2, pp. 210-220, 2011.
- [6] M. Kristan, S. Kovačič, A. Leonardis, et al. "A two-stage dynamic model for visual tracking," *IEEE Systems, Man, and Cybernetics, Part B*, 40(6), 2010, p. 1505-1520.
- [7] J. Xiao, R. Stolkin, M. Oussalah, A. Leonardis, "Continuously adaptive data fusion and model re-learning for particle filter tracking with multiple features", *IEEE Sensors*, 99, 2016, DOI: 10.1109/ISEN.2016.2514704
- [8] O.-D. Nouar, G. Ali, and C. Raphael, "Improved object tracking with camshift algorithm," in *Proc. International Conf. Acoustics, Speech and Signal Processing*, vol. 2. IEEE, 2006, pp. II-II.
- [9] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 564-577, 2003.
- [10] S.-K. Weng, C.-M. Kuo, and S.-K. Tu, "Video object tracking using adaptive kalman filter," *Journal of Visual Communication and Image Representation*, vol. 17, no. 6, pp. 1190-1208, 2006.
- [11] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, no. 1, pp. 99-110, 2003.
- [12] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174-188, 2002.
- [13] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 146, no. 1, pp. 2-7, 1999.
- [14] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. European Conf. Computer Vision*. Springer, pp. 661-675, 2002.
- [15] J. Li and C.-S. Chua, "Transductive local exploration particle filter for object tracking," *Image and Vision Computing*, vol. 25, no. 5, pp. 544-552, 2007.
- [16] J. Xiao and M. Oussalah, "Scale modification through particle distribution in color based tracking," in *Proc. European Conf. Visual Media Production*, 2013.
- [17] L. Čehovin, M. Kristan and A. Leonardis, "Robust Visual Tracking Using an Adaptive Coupled-Layer Visual Model," in *PAMI*, vol.35, no.4, pp.941-953, 2013.
- [18] S. Hare, A. Saffari, and P. H. S. Torr. "Struck: Structured Output Tracking with Kernels." in *ICCV*, 2011.
- [19] X. Mei and H. Ling. "Robust visual tracking using l1 minimization." in *CVPR*, pp. 1436-1443, 2009.
- [20] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the Circulant Structure of Tracking-by-detection with Kernels," in *ECCV*, 2012.
- [21] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. "Incremental Learning for Robust Visual Tracking." in *IJCV*, pp:125-141, 2008.
- [22] M. Talha and R. Stolkin, "Particle filter tracking of camouflaged targets by adaptive fusion of thermal and visible spectra camera data," *Sensors Journal, IEEE transactions on*, vol. 14, no. 1, pp. 159-166, 2014.
- [23] J. Xiao and M. Oussalah, Collaborative Tracking for Multiple Objects in the presence of inter-occlusions, *IEEE Transactions on Circuits and Systems for Video Technology*, 01, 2015, DOI: 10.1109/TCSVT.2015.2406193
- [24] J. Wang, X. Chen, and W. Gao, "Online selecting discriminative tracking features using particle filter," in *Proc. International Conf. Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2005, pp. 1037-1042.
- [25] R. Stolkin, D. Rees, M. Talha, and I. Florescu, "Bayesian fusion of thermal and visible spectra camera data for region based tracking with rapid background adaptation," in *Proc. International Conf. Multisensor Fusion and Integration for Intelligent Systems*. IEEE, 2012, pp. 192-199.
- [26] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197-208, 2000.
- [27] A. Bhattacharyya, "On a Measure of Divergence between Two Multinomial Populations," *Sankhya*, vol. 7, pp. 401-406, 1946.
- [28] R. T. Collins, "Mean-shift blob tracking through scale space," in *Proc. International Conf. Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2003, pp. 234-240.
- [29] M. Kristan, et al. "The visual object tracking vot2013 challenge results." in *ICCVW*, pp. 98-111, 2013.
- [30] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark." in *CVPR*, pp. 2411-2418, 2013.