DEEP CONVOLUTIONAL PARTICLE FILTER WITH ADAPTIVE CORRELATION MAPS FOR VISUAL TRACKING

Reza Jalil Mozhdehi, Yevgeniy Reznichenko, Abubakar Siddique and Henry Medeiros

{reza.jalilmozhdehi, yevgeniy.reznichenko, abubakar.siddique and henry.medeiros}@marquette.edu
Electrical and Computer Engineering Department, Marquette University, Milwaukee, WI, USA

ABSTRACT

The robustness of the visual trackers based on the correlation maps generated from convolutional neural networks can be substantially improved if these maps are used to employed in conjunction with a particle filter. In this article, we present a particle filter that estimates the target size as well as the target position and that utilizes a new adaptive correlation filter to account for potential errors in the model generation. Thus, instead of generating one model which is highly dependent on the estimated target position and size, we generate a variable number of target models based on high likelihood particles, which increases in challenging situations and decreases in less complex scenarios. Experimental results on the Visual Tracker Benchmark v1.0 demonstrate that our proposed framework significantly outperforms state-of-the-art methods.

***Index Terms*—** Particle Filter, Target Model, Correlation Map, Deep Convolutional Neural Network, Visual Tracking

1. INTRODUCTION

One important aspect of visual target tracking is to accurately determine the target position as well as its size. Particle filters have been employed in visual tracking for years because of their ability to perform these tasks robustly [1]. Recently, deep convolutional neural networks have been used to produce highly discriminative target features [2]. Trackers such as HCFT and HDT [3, 4], which employ convolutional features in conjunction with correlation filters have shown better visual tracking performance than traditional trackers such as MEEM [5], KCF [6], Struck [7], SCM [8] and TLD [9]. Despite the substantial performance gains obtained in recent years by the aforementioned CNN-correlation methods, their main disadvantage is their inability to vary the size of the target bounding box [3, 4]. By combining CNN-based correlation maps with particle filters, we can overcome this limitation and improve overall tracking accuracy and robustness.

We propose two new mechanisms to improve the performance of our previous visual tracker named Deep Convolutional Particle Filter (DCPF) [10]. Briefly, DCPF uses multiple particles as inputs to VGG-Net [2]. For each particle, it then applies the correlation filter used in HCFT on the extracted hierarchical convolutional features to construct the correlation map. The target position at the current frame is calculated based on the response of the correlation maps. However, similar to HCFT, DCPF tracks a bounding box of fixed size. Another limitation of DCPF and other trackers based on conventional correlation filters is the fact that they generate only one target model. Thus, errors in calculating the final target state cause the target model to be incorrectly updated. In our new visual tracker named DCPF2, we address the first limitation by extending DCPF's particle filter to estimate the target size. In addition, we employ an adaptive correlation filter, which produces a variable number of target models based on the number of high-likelihood particles that have been generated in the previous frame. In frames where the target can be easily tracked, this number is low because the best particle has a high likelihood and fewer particles have similar likelihoods. Conversely, in challenging situations, the target is less similar to the model and hence the likelihood of most particles decreases and the particle weight distribution becomes less centralized. We used the Visual Tracker Benchmark v1.0 [11] to evaluate the performance of the proposed tracker and showed that it performs favorably with respect to other state-of-the-art methods.

2. PROPOSED ALGORITHM

In this section, we first explain how our particle filter estimates the target size and its position. Then, our adaptive correction filter is discussed. Fig. 1 illustrates the proposed approach.

2.1. Particle Filter to Estimate the Target Bounding Box

Let the target position and size be represented by

$$z_t = \left[u_t, v_t, h_t, w_t\right]^T, \tag{1}$$

where u_t and v_t are the locations of the target on the horizontal and vertical image axes at frame t, and h_t and w_t are its

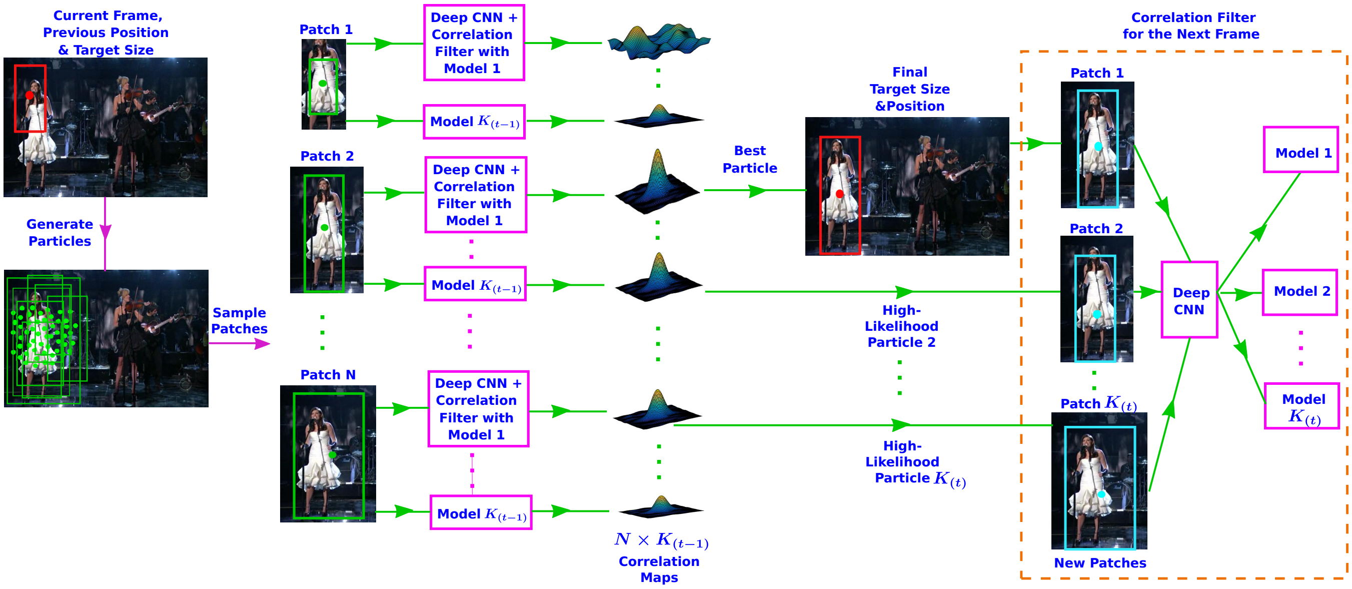
Current Frame, Previous Position & Target Size
Generate Particles
Sample Patches
Patch 1
Patch 2
Patch N
Deep CNN + Correlation Filter with Model 1
Model $K_{(t-1)}$
$N \times K_{(t-1)}$ Correlation Maps
Best Particle
Final Target Size &Position
High-Likelihood Particle 2
High-Likelihood Particle $K_{(t)}$
Correlation Filter for the Next Frame
Patch 1
Patch 2
Patch $K_{(t)}$
New Patches
Deep CNN
Model 1
Model 2
Model $K_{(t)}$

Fig. 1. Overview of our proposed tracker.

width and height. The target state is given by

$$x_t = \left[z_t, \dot{z}_t\right]^T, \tag{2}$$

where $\dot{z}_t$ is the velocity of z_t. The tracker employs a linear motion model to predict the current state of the target $\hat{x}_t$ based on the previous target state x_{t-1}. The predicted target state is given by

$$\hat{x}_t = Ax_{t-1}, \tag{3}$$

where A is a standard constant velocity process matrix. Then, particles $x^{(i)} = \left[z^{(i)}, \dot{z}^{(i)}\right]^T$ are generated by adding samples $\eta^{(i)} \in \mathbb{R}^8$ drawn from a zero-mean normal distribution. That is,

$$x_t^{(i)} = \hat{x}_t + \eta^{(i)}, \tag{4}$$

where $i = 1, ..., N$ and N is the number of the particles. In order to limit the number of particles needed, rather than drawing $\eta^{(i)}$ directly from an eight-dimensional distribution, we draw its samples individually, and change its height and width simultaneously using the same sample (i.e., we change the target scale but not its aspect ratio).

In the next step, $z^{(i)}$ are used to sample different patches from frame t. Each patch is then fed into the CNN to calculate its convolutional feature map. Let $f_{l,d}^{(i)} \in \mathbb{R}^{M\times Q}$ be the convolutional feature map, where M, Q are the width and height of the map, l is the convolutional layer and d is the number of the channels for that layer $d = 1, ..., D$. Then, its correlation response map $R_l^{(j)(i)} \in \mathbb{R}^{M\times Q}$ is given by

$$R_l^{(j)(i)} = \mathfrak{F}^{-1}(\sum_{d=1}^{D} C_{l,d}^{(j)} \odot \bar{f}_{l,d}^{(i)}), \tag{5}$$

where $\mathfrak{F}$ represents the inverse Fourier transform [3], $j = 1, ..., K_{t-1}$ illustrates the number of models generated in the previous frame $t-1$, $C_{l,d}^{(j)}$ represents channel d of layer l of the correlation filter of the model j, the bar represents complex conjugation and $\odot$ is the Hadamard product. The final correlation response map $R^{*(j)(i)}$ for particle i and model j is calculated based on a weighted sum of the maps for all the CNN layers as proposed in [3]. The likelihood or weight of each correlation response map is calculated by

$$\omega^{(j)(i)} = \sum_{m=1}^{M} \sum_{q=1}^{Q} R_{(m,q)}^{*(j)(i)}, \tag{6}$$

where $R_{(m,q)}^{*(j)(i)}$ refers to the element of the final correlation response map on row m and column q. In total, we have $N \times K_{t-1}$ weights. The intuition behind this choice is that feature maps that correspond to the target tend to show substantially higher correlation values than background patches [12]. We then find the maximum weight ω_{max} over all the particles and models

$$\omega_{max} = \max_{j,i} \omega^{(i,j)}. \tag{7}$$

Let the indexes corresponding to ω_{max} be $i = i^*$ (the best particle) and $j = j^*$ (the best model). Then, the final target size is given by $h^{(i^*)}$ and $w^{(i^*)}$. That is, the i^*-th patch with dimensions $h^{(i^*)}$ and $w^{(i^*)}$ is the most similar to the best model $C^{(j^*)}$. Additionally, let $R^{*(j^*)(i^*)}$ be the correlation response map associated with ω_{max}, its peak is located at

$$[\delta_u, \delta_v] = \operatorname*{argmax}_{m,q} R_{(m,q)}^{*(j^*)(i^*)}, \tag{8}$$

where $m = 1, ..., M$ and $q = 1, ..., Q$. The final target position is then calculated by shifting the best particle towards the peak of its correlation map

$$[\tilde{u}, \tilde{v}] = [u^{(i^*)} + \delta_u, v^{(i^*)} + \delta_v], \tag{9}$$

where $u^{(i^*)}$ and $v^{(i^*)}$ correspond to the position of the best particle. Thus, the target state at the frame t is

$$x_t = \left[z^*, \dot{z}^{(i^*)}\right]^T, \tag{10}$$

where $\dot{z}^{(i^*)}$ is the velocity of the best particle and

$$z^* = \left[\tilde{u}, \tilde{v}, h^{(i^*)}, w^{(i^*)}\right]^T. \tag{11}$$

Algorithm 1 summarizes our method to estimate the target state.

```
Algorithm 1 Calculate the current target state
Input: Current frame, previous target state $x_{t-1}$, correlation
       filters $C^{(j)}, j = 1, \ldots, K_{t-1}$ generated in the
       previous frame
Output: Current target state $x_t$, maximum weight $\omega_{max}$,
        $N$ particles $x^{(i)}$, $N \times K_{t-1}$ correlation response
        maps $R^{*(j)(i)}$ and their weights $\omega^{(j)(i)}$
Generate $N$ particles around the predicted target state
    $\hat{x}_t$ according to Eqs. 1 to 4
for Each particle $x^{(i)}$ do
    for Each of the $K_{t-1}$ correlation filters $C^{(j)}$ do
        Generate the $K_{t-1}$ correlation response
            maps $R^{*(j)(i)}$ according to Eq. 5
        Compute the weight $\omega^{(j)(i)}$ based on Eq. 6
    end for
end for
Determine the best particle using Eq. 8
Update the target state $x_t$ according to Eqs. 9 to 11
```

2.2. Adaptive Correlation Filter

After finding ω_{max}, we examine the following relationship for all $N \times K_{t-1}$ weights

$$\omega^{(j)(i)} > \alpha\omega_{max}, \tag{12}$$

where α is a constant. If Eq. 12 is true, the corresponding particle is considered a high likelihood particle. Let i' and j' be the indices of the selected weights. Then, $h^{(i')}$ and $w^{(i')}$ calculated by Eq. 4 are considered the target size particle. Additionally, its associated correlation response map $R^{*(j')(i')}$ is used to calculate the estimated target position $\tilde{u}^{i'}$ and $\tilde{v}^{i'}$ similar to Eq. 8 and Eq. 9. Thus the corresponding high likelihood particle $z^{(s)}_{high}$ is given by

$$z^{(s)}_{high} = \left[\tilde{u}^{i'}, \tilde{v}^{i'}, h^{(i')}, w^{(i')}\right]^T, \tag{13}$$

where $s = 1, ..., K_t$ and K_t is the number of the high-likelihood particles. We then generate a patch from frame t for each of the K_t high-likelihood particles. In the next step, these patches are fed into the CNN to extract K_t convolutional feature maps, and a new correlation filter $C^{(s)}_{l,d}$ is then generated for each of the K_t high likelihood particles. The generated models are used in frame $t+1$ to be compared with the convolutional features generated by each particle.

```
Algorithm 2 Adaptive Correlation Filter
Input: Current frame, maximum weight $\omega_{max}$, $N$ particles
       $x^{(i)}$, their $N \times K_{t-1}$ correlation response maps
       $R^{*(j)(i)}$ and weights $\omega^{(j)(i)}$
Output: Correlation filters $C^{(j)}, j = 1, \ldots, K_t$ to be used in
        the next frame
for Each weight $\omega_i^j$ do
    if Eq. 12 is true then
        Generate a high-likelihood particle $z^{(s)}_{high}$
            according to Eq. 13
    end if
end for
for Each particle $z^{(s)}_{high}$ do
    Calculate its correlation filter $C^{(s)}_{l,d}$
end for
```

Alg. 2 summarizes our adaptive correlation filter procedure. The comparison between the best model with the most accurate target size and position generates more accurate correlation maps. As previously mentioned, by varying the number of models K_t with the number of high likelihood particles, we are able to maintain a larger number of tentative models in challenging scenarios such as in the presence of illumination variation, motion blur, or partial occlusion due to the wider distribution of the particle weights under these conditions.

3. RESULTS AND DISCUSSION

We evaluate our tracker on the well known Visual Tracker Benchmark v1.0 [11]. This benchmark contains 50 data sequences that are annotated with 9 attributes representing challenging aspects of tracking, such as scale variation, in-plane rotation and illumination variations. We use a one-pass evaluation (OPE) in which the tracker is initialized with the ground truth location at the first frame of the image sequence [11]. Also, we heuristically set $\alpha = 0.8$ and $N = 300$.

Fig. 2 qualitatively illustrates the performance of our tracker in comparison with three trackers: the CNN-based trackers HCFT and HDT as well as the correlation filter tracker SCT6 [13]. As the results in Fig. 2 indicate, the baseline trackers get easily confused in situations such as scale variation, illumination variation, in-plane and out-of-plane rotations. The proposed particle-correlation filter is able to sample several image patches and it is hence capable

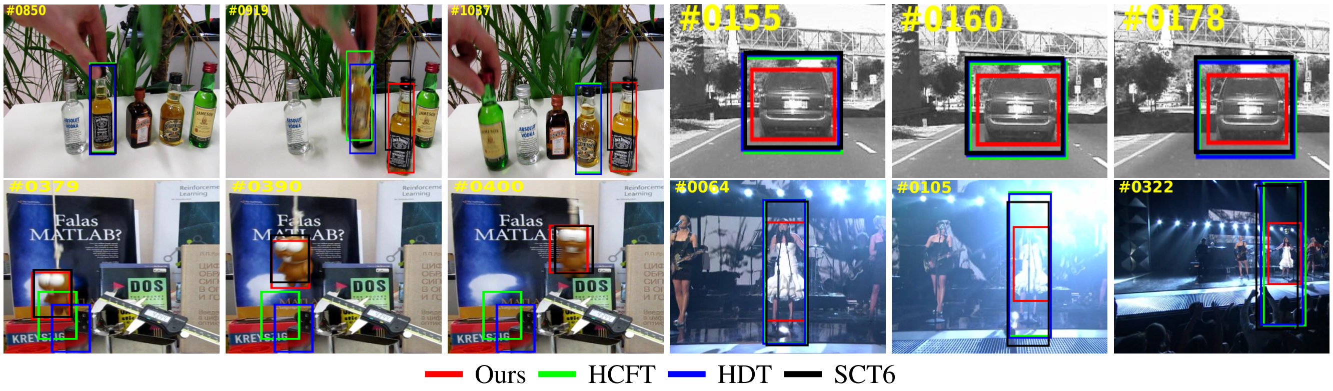

Fig. 2. Qualitative evaluation of our tracker, *HCFT*, *HDT* and *SCT*6 on six challenging sequences (from left to right and top to bottom are *Liquor*, *Car4*, *Lemming* and *Singer1*, respectively).

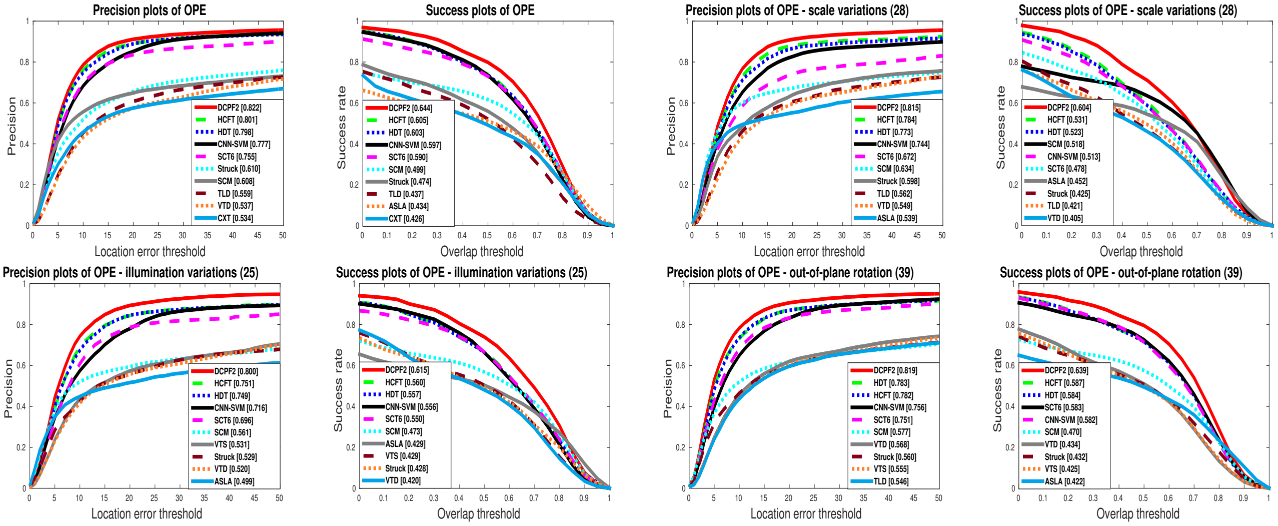

Fig. 3. Quantitative evaluation of our tracker and fourteen state-of-the-art trackers on OPE.

of overcoming these difficulties.

Fig. 3 provides a quantitative evaluation of our proposed approach in comparison with 11 state-of-the-art trackers [3, 4, 14, 13, 7, 8, 9, 15, 16, 17, 18]. In the figure, *Precision plots* correspond to the average Euclidean distance between the tracked locations and the ground truth while *Success plots* correspond to the area of overlap between the predicted bounding box and the respective ground truth [11]. In attributes such as scale variation, illumination and out-of-plane rotations where the common correlation filter loses track of the target, our adaptive correlation filter in conjunction with the particle filter are then able to recover using the weights generated by the CNN. For challenging scenarios of scale variation, illumination variation and out-of-plane rotation as well in terms of overall performance, our tracker shows improvements of approximately 14%, 10%, 9% and 7%, respectively, in comparison with the second best tracker HCFT.

4. CONCLUSION

This article proposes a novel framework named DCPF2 for visual tracking. We extend the particle filter employed in our previous tracker DCPF to estimate the target size. Additionally, because the correlation filter used in DCPF is heavily dependent upon the estimated target position, we find all of the high-likelihood particles and calculate a model for each of them. The Visual Tracker Benchmark v1.0 is used for evaluating the proposed tracker's performance. The results show that these strategies improve the performance of CNN-correlation trackers in critical situations such as scale variation, illumination and out-of-plane rotations.

5. REFERENCES

[1] A. D. Bimbo and F. Dini, "Particle filter-based visual tracking with a first order dynamic model and uncer-

tainty adaptation," *Computer Vision and Image Understanding*, vol. 115, no. 6, pp. 771 – 786, 2011.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, May 2015.

[3] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[4] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M. H. Yang, "Hedged deep tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 4303–4311.

[5] J. Zhang, S. Ma, and S. Sclaroff, "Robust tracking via multiple experts," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 188–203.

[6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[7] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *IEEE International Conference on Computer Vision (ICCV)*. 2011, pp. 263–270, IEEE Computer Society.

[8] W. Zhong, H. Lu, and M. H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2356–2368, 2014.

[9] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[10] R. J. Mozhdehi and H. Medeiros, "Deep convolutional particle filter for visual tracking," in *IEEE International Conference on Image Processing (ICIP)*, 2017.

[11] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[12] R. Walsh and H. Medeiros, "Detecting tracking failures from correlation response maps," in *Advances in Visual Computing: 12th International Symposium (ISVC)*, 2016, pp. 125–135.

[13] J. Choi, H. J. Chang, J. Jeong, Y. Demiris, and J. Y. Choi, "Visual tracking using attention-modulated disintegration and integration," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *32nd International Conference on Machine Learning*, 2015.

[15] X. Jia, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012, pp. 1822–1829, IEEE Computer Society.

[16] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1269–1276.

[17] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[18] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 1195–1202.