

Deep Convolutional Likelihood Particle Filter for Visual Tracking

Reza Jalil Mozhdehi and Henry Medeiros

Marquette University, Milwaukee, WI, USA,

reza.jalilmozhdehi@marquette.edu and henry.medeiros@marquette.edu

Abstract. We propose a novel particle filter for convolutional-correlation visual trackers. Our method uses correlation response maps to estimate likelihood distributions and employs these likelihoods as proposal densities to sample particles. Likelihood distributions are more reliable than proposal densities based on target transition distributions because correlation response maps provide additional information regarding the target’s location. Additionally, our particle filter searches for multiple modes in the likelihood distribution, which improves performance in target occlusion scenarios while decreasing computational costs by more efficiently sampling particles. In other challenging scenarios such as those involving motion blur, where only one mode is present but a larger search area may be necessary, our particle filter allows for the variance of the likelihood distribution to increase. We tested our algorithm on the Visual Tracker Benchmark v1.1 (OTB100) and our experimental results demonstrate that our framework outperforms state-of-the-art methods.

Keywords: Likelihood Particle Filter, Gaussian Mixture Model, Deep Convolutional Neural Network, Correlation Response Map, Visual Tracking.

1 Introduction

Particle filters are widely applied in visual tracking problems due to their ability to find targets in challenging scenarios such as those involving occlusions or fast motion. Recently, particle filters have been used in conjunction with deep convolutional neural networks (CNN) [6,12] and correlation filters [2,15,11,7]. The Hierarchical Convolutional Feature Tracker (HCFT) proposed by Ma et al. in [7] showed significant performance improvements over previous works, demonstrating the effectiveness of using convolutional features along with correlation filters. Correlation filters provide a map showing similarities between convolutional features corresponding to an image patch and the target [2,15,3]. Adding a particle filter to convolutional-correlation visual trackers can significantly improve their results as shown in [16,14,8,10,9]. In these methods, particle filters sample several image patches and calculate the weight of each sample by applying a correlation filter to the convolutional response maps.

In this work, we propose a novel convolutional-correlation particle filter for visual tracking which estimates likelihood distributions from correlation response maps. Sampling particles from likelihood distributions improves the accuracy of patch candidates because correlation response maps have an initial evaluation of the target location. Thus,

they are more reliable proposal densities than transition distributions, commonly used in particle-correlation trackers such as [16,14,8,10]. Additionally, these trackers calculate the posterior distribution based on the peaks of correlation maps without considering them in the computation of particle weights. Our particle filter solves this problem using a multi-modal likelihood distribution to address challenging tracking scenarios. Our proposed algorithm also calculates a likelihood distribution with larger variances, which is useful in other challenging scenarios involving fast motion or background clutter because it expands the target search area. Additionally, this method decreases the number of required particles. Experimental results on the Visual Tracker Benchmark v1.1 (OTB100) [13] show that our proposed framework outperforms state-of-the-art methods.

2 The change of support problem in convolution-correlation particle filters

The particle weights in a particle filter are calculated by [1]

$$\omega_{x_t}^{(i)} \propto \omega_{x_{t-1}}^{(i)} \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1})}{q(x_t^{(i)}|x_{t-1}, y_t)}, \quad (1)$$

where $p(x_t^{(i)}|x_{t-1})$ and $p(y_t|x_t^{(i)})$ are the transition and likelihood distributions, and $q(x_t^{(i)}|x_{t-1}, y_t)$ is the proposal distribution used to sample the particles. The posterior distribution is then approximated by

$$\hat{Pr}(x_t|y_t) \approx \sum_{i=1}^N \varpi_{x_t}^{(i)} \delta(x_t - x_t^{(i)}), \quad (2)$$

where $\varpi_{x_t}^{(i)}$ are the normalized weights. However, particle filters used in correlation trackers generally sample particles from the transition distribution, i.e., $q(x_t^{(i)}|x_{t-1}, y_t) = p(x_t^{(i)}|x_{t-1})$. These methods also re-sample particles at every frame, which removes the term corresponding to previous weights $\omega_{x_{t-1}}^{(i)}$ from Eq. 1. Finally, the weight of each particle in these trackers is given by [16]

$$\omega_{x_t}^{(i)} \propto p(y_t|x_t^{(i)}), \quad (3)$$

where $p(y_t|x_t^{(i)})$ is a function of $R_{x_t^{(i)}}^{y_t} \in \mathbb{R}^{M \times Q}$, the correlation response map centered at $x_t^{(i)}$. In these trackers, particles are shifted to the peaks of correlation maps and the posterior distribution is then approximated by the particles' weights at the shifted locations, i.e.,

$$\hat{Pr}(x_t|y_t) \approx \sum_{i=1}^N \varpi_{x_t}^{(i)} \delta(x_t - \tilde{x}_t^{(i)}), \quad (4)$$

where $\tilde{x}_t^{(i)}$ is the peak of the correlation response map corresponding to the i -th particle. However, the posterior distribution using the shifted locations must consider the

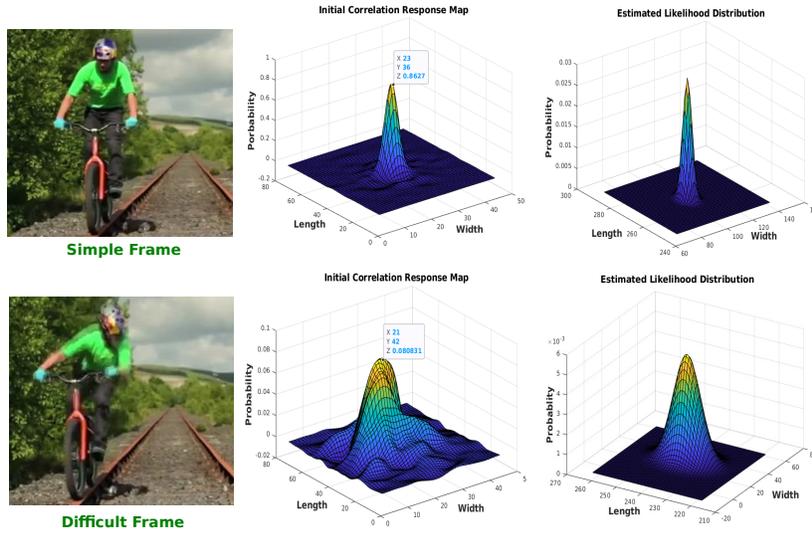


Fig. 1. Estimated likelihood distributions for common scenarios (simple frame) and a challenging scenario involving fast motion (difficult frame).

weights corresponding to the new support points, not the original locations of the particles. That is, the original locations are used in weight computation, but the shifted support is used to approximate the posterior distribution. To solve this, we sample particles from the likelihood distribution instead. Particle filters that sample from likelihood distributions generate more accurate particles, but sampling from the likelihood distribution is not always possible. Fortunately, convolutional-correlation trackers generate correlation maps that can be used in the construction of likelihood distributions.

3 Likelihood Particle Filter

Our algorithm generates an initial correlation response map for the current frame based on the previously estimated target state to calculate an initial likelihood distribution. That is, we generate a patch from the current frame based on the previous target state and use a CNN [12] to extract the convolutional features from this patch. We then compare these features with the target model to calculate the final correlation response map [7]. As seen in Fig. 1, in most scenarios (which we call “simple frames”) the correlation response map corresponds to a sharp Gaussian distribution with a prominent peak. In challenging scenarios (“difficult frames”), correlation maps are wider with less pronounced peaks. We need to estimate likelihood distributions consistently in both scenarios. To address this issue, we fit a Gaussian distribution to the correlation response maps while disregarding elements with probability lower than a threshold τ . By disregarding low probability elements, we mitigate the impact of the background on the

computation of the model. We compute the mean of the correlation response map using

$$\mu \approx \frac{\sum_{i=1}^u q_i s_i}{\sum_{i=1}^u q_i}, \quad (5)$$

where s_i and q_i represent the elements of the correlation response map and their respective probabilities, and u is the number of elements with probability higher than τ . The variance of the response map is then given by

$$\sigma^2 \approx \frac{\sum_{i=1}^u q_i (s_i - \mu)^2}{\sum_{i=1}^u q_i}. \quad (6)$$

Thus, our model assigns low probabilities to pixels that are likely to belong to the background while assigning relatively high probabilities to all the regions that might correspond to the target. As a result, our samples concentrate in regions where the target is more likely to be present.

Fig. 1 shows our estimated likelihood distributions for two different frames of the *Biker* data sequence of the OTB100 benchmark. In the difficult frame, the target undergoes motion blur, which causes the correlation response map to be wider with a lower peak. Our estimated variance is then correspondingly higher, which helps our tracker to sample particles over a wider area to compensate for tracking uncertainties in difficult scenarios. The example in Fig. 2 shows how the variance increases as the target approaches difficult frames.

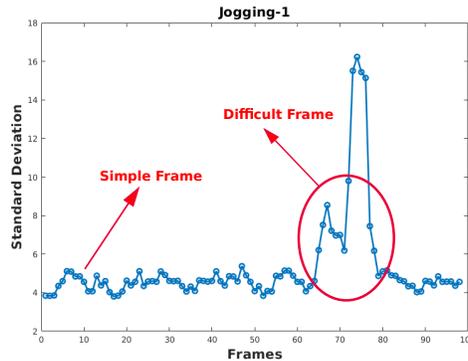


Fig. 2. Standard deviations of the estimated likelihood distributions in data sequence *Jogging-1* of the OTB-100 dataset.

Although allowing for higher variances in challenging scenarios such as those involving fast motion helps our tracker address such issues, this strategy alone cannot handle multi-modal correlation response maps. To resolve this issue, we propose to determine the peaks of the distribution using the approach described below.

3.1 Multi-modal likelihood estimation

The existence of multiple peaks in a correlation response map usually indicates the presence of confusing elements in the background of the frame, as the example in Fig. 3 illustrates. In the frame shown in the figure, there are two peaks in the correlation response map when partial target occlusion occurs. The peaks correspond to the woman on the left side of the image (the target) and the pole partially occluding her. By applying a threshold to remove low probability elements from the correlation response map, two clusters become apparent.

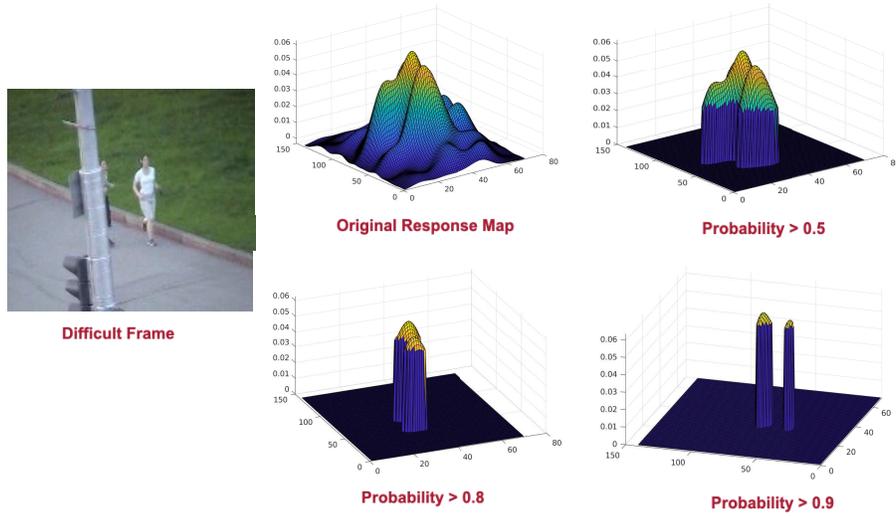


Fig. 3. A difficult frame including target occlusion. Its correlation response map has two peaks. By increasing the threshold to remove low probability elements, two clusters corresponding to the target and the pole are seen.

To identify the peaks of the correlation map while disregarding additional background clutter, we remove from the map points with probability lower than a threshold τ . We then fit a Gaussian mixture model to the remaining feature map points which clusters them into k groups [5]. Fig. 4 shows two instances of correlation response maps in which we identify $k = 2$ and $k = 3$ clusters. The likelihood corresponding to each peak is then given by a normal distribution with mean and variance given by Eqs. 5 and 6. Algorithm 1 summarizes our proposed approach to estimate the likelihood distribution for each cluster.

3.2 Particle sampling

We sample particles from the Gaussian likelihood distributions obtained from the correlation response maps in the current frame. The probability that a particle is sampled

Algorithm 1 Multi-modal likelihood estimation.**Input:** Current frame y_t and previous target state x_{t-1} **Output:** One likelihood distribution for each correlation map cluster

- 1: Extract a patch from the current frame based on the previous target state
- 2: Extract the CNN features of the patch and calculate its correlation response map
- 3: Remove points with probability lower than τ
- 4: Fit a Gaussian mixture model to the map and find the clusters
- 5: Estimate the likelihood distribution of each cluster based on the mean and variance of its elements in the map according to Eqs. 5 and 6

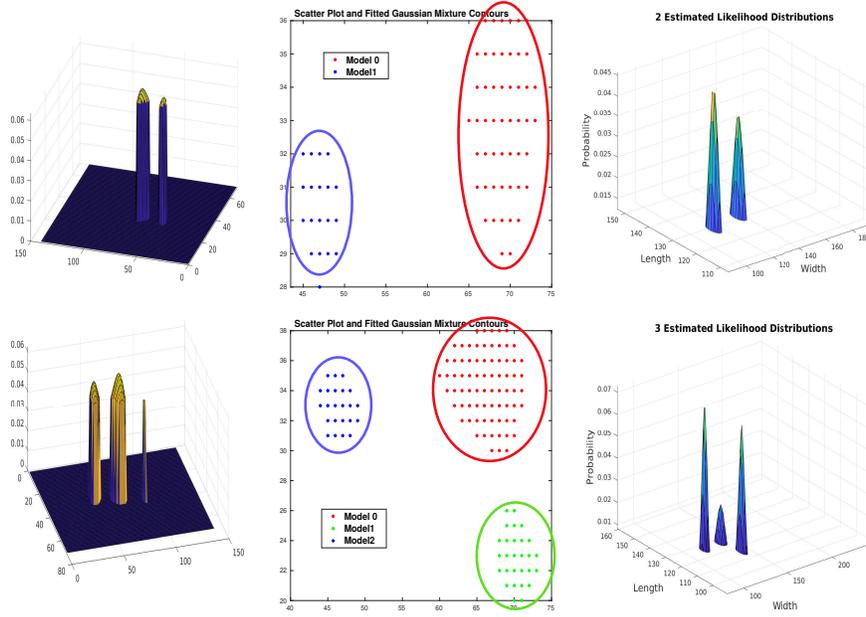


Fig. 4. Finding clusters; left: correlation response maps with two and three clusters, middle: clusters of the correlation response maps obtained by fitting a Gaussian mixture model, right: estimated likelihood distributions for each cluster.

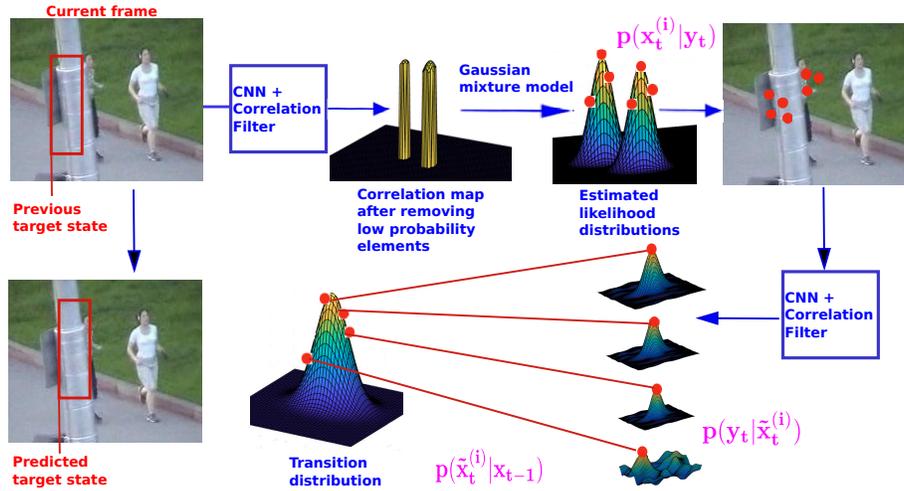


Fig. 5. Overview of the steps comprising the proposed DCPF-Likelihood visual tracker.

from the likelihood distribution is given by

$$p(x_t^{(i)} | y_t) \propto \sum_{j=1}^k \mathcal{N}(x_t^{(i)}; \mu_j, \sigma_j), \quad (7)$$

where μ_j and σ_j are the mean and variance of the j -th mode of the likelihood. We generate a patch for each particle and extract its features using a CNN. After calculating the correlation response map for each particle, we shift the particles to the peaks of their respective correlation response maps. The peak of each correlation response map is the estimated target position based on the patch centered at the corresponding particle. Because each particle is shifted to the peak of the correlation response map, we consider $p(\tilde{x}_t^{(i)} | x_t^{(i)}) = 1$, where $\tilde{x}_t^{(i)}$ is the peak of the corresponding correlation response map. As a result, $p(x_t^{(i)} | y_t) = p(\tilde{x}_t^{(i)} | y_t)$.

3.3 Calculating the weights and posterior distribution

By computing the weight of each shifted particle $\tilde{x}_t^{(i)}$, we can accurately estimate the posterior based on the shifted particles and their correct weights, which addresses the problem of incorrect support points observed in previous works. As discussed earlier, Eq. 1 corresponds to the weight of each particle before shifting. The weight of the shifted particles is then given by

$$\omega_{\tilde{x}_t^{(i)}} \propto \omega_{x_{t-1}^{(i)}} \frac{p(y_t | \tilde{x}_t^{(i)}) p(\tilde{x}_t^{(i)} | x_{t-1}^{(i)})}{q(\tilde{x}_t^{(i)} | x_{t-1}^{(i)}, y_t)}, \quad (8)$$

where the term corresponding to the previous weight is removed because we perform resampling at every frame. Additionally, [1]

$$q(\tilde{x}_t^{(i)} | x_{t-1}, y_t) = p(\tilde{x}_t^{(i)} | y_t). \quad (9)$$

Thus, the weight of each shifted particle is

$$\omega_{\tilde{x}_t}^{(i)} \propto \frac{p(y_t | \tilde{x}_t^{(i)}) p(\tilde{x}_t^{(i)} | x_{t-1})}{p(\tilde{x}_t^{(i)} | y_t)}. \quad (10)$$

Let the target state be defined as

$$z_{t-1} = [x_{t-1}, \dot{x}_{t-1}]^T, \quad (11)$$

where \dot{x}_{t-1} is the velocity of x_{t-1} . We apply a first-order motion model to z_{t-1} according to

$$\bar{z}_{t-1} = Az_{t-1}, \quad (12)$$

where \bar{z}_{t-1} represents the predicted target state and A is the process matrix defined by

$$A = \begin{bmatrix} I_4 & | & I_4 \\ \hline 0_{(4,4)} & | & I_4 \end{bmatrix}, \quad (13)$$

where I_4 is a 4×4 identity matrix and $0_{(4,4)}$ is a 4×4 zero matrix. We use a Gaussian distribution $\mathcal{N}(\bar{x}_{t-1}, \sigma^2)$ to find the probability of each estimated particle in the current frame $p(\tilde{x}_t^{(i)} | x_{t-1})$.

Additionally, $p(y_t | \tilde{x}_t^{(i)})$ is the likelihood of each shifted particle. Let $f_{x_t^{(i)}}(l, o)$ be the convolutional features of each particle $x_t^{(i)}$ where l and o represent the layers and channels of the network, respectively. The correlation response map is then calculated by [7]

$$R_{x_t^{(i)}}^{y_t}(x) = \sum_{l=1}^L \Upsilon_l(\mathfrak{F}^{-1}(\sum_{o=1}^O C_{t-1}(l, o) \odot \bar{F}_{x_t^{(i)}}(l, o))), \quad (14)$$

where $\bar{F}_{x_t^{(i)}}(l, o)$ is the complex conjugate Fourier transform of $f_{x_t^{(i)}}(l, o)$, C_{t-1} is the model generated in the previous frame, \odot represents the Hadamard product, \mathfrak{F}^{-1} is the inverse Fourier transform operator, and Υ_l is a regularization term [7]. The peak of $R_{x_t^{(i)}}^{y_t}$ is then calculated by

$$\tilde{x}_t^{(i)} = \arg \max_{m,q} R_{x_t^{(i)}}^{y_t}(m, q), \quad (15)$$

where $m = 1, \dots, M$ and $q = 1, \dots, Q$. The likelihood of $\tilde{x}_t^{(i)}$ is calculated by [10]

$$p(y_t | \tilde{x}_t^{(i)}) = \frac{1}{M \times Q} \sum_{m,q} R_{\tilde{x}_t^{(i)}}^{y_t}(m, q). \quad (16)$$

The posterior distribution based on the shifted particles and their respective weights is then

$$\hat{Pr}(x_t | y_t) \approx \sum_{i=1}^N \varpi_{\tilde{x}_t}^{(i)} \delta(x_t - \tilde{x}_t^{(i)}), \quad (17)$$

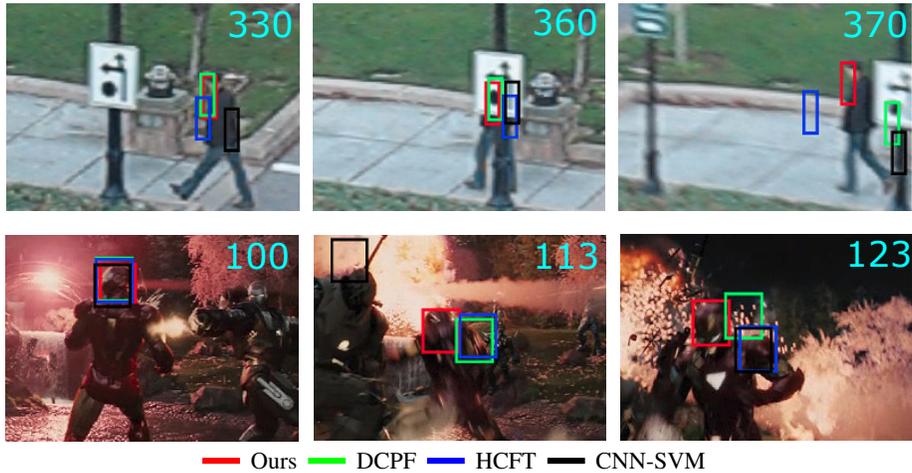


Fig. 6. Qualitative evaluation of our tracker against *DCPF*, *HCFT*, and *CNN-SVM* on two challenging sequences: *Human6* (top) and *Ironman* (bottom).

where $\varpi_{\hat{x}_t}^{(i)}$ is the normalized version of $\omega_{\hat{x}_t}^{(i)}$. Fig. 5 summarizes the steps of our method, and Algorithm 2 describes the details of our approach.

Algorithm 2 DCPF-Likelihood visual tracker.

Input: Current frame y_t and previous target state x_{t-1}

Output: Current target state x_t

- 1: Estimate a likelihood distribution for each cluster using Algorithm 1
 - 2: Sample particles from the likelihood distributions $p(x_t^{(i)} | y_t)$
 - 3: Extract the CNN features of the patches corresponding to each particle and calculate its correlation response map according to Eq. 14
 - 4: Shift the particles to the peaks of their correlation response maps based on Eq. 15
 - 5: Calculate the likelihood $p(y_t | \hat{x}_t^{(i)})$ based on Eq. 16
 - 6: Calculate the transition probability $p(\hat{x}_t^{(i)} | x_{t-1})$ according to Eqs. 11 to 13
 - 7: Compute the weight of each shifted particle $\omega_{\hat{x}_t}^{(i)}$ according to Eqs. 8 to 10
 - 8: Calculate the posterior distribution according to Eq. 17
-

4 Experimental results

We use the Visual Tracker Benchmark v1.1 (OTB100) to assess the performance of our tracker. This benchmark contains 100 video sequences, which include 11 challenging scenarios. Our results are based on the one-pass evaluation (OPE), which uses the ground truth target size and position in the first frame to initialize the tracker. Our evaluation is based on the precision and success measures, described in [13]. Fig. 6 shows

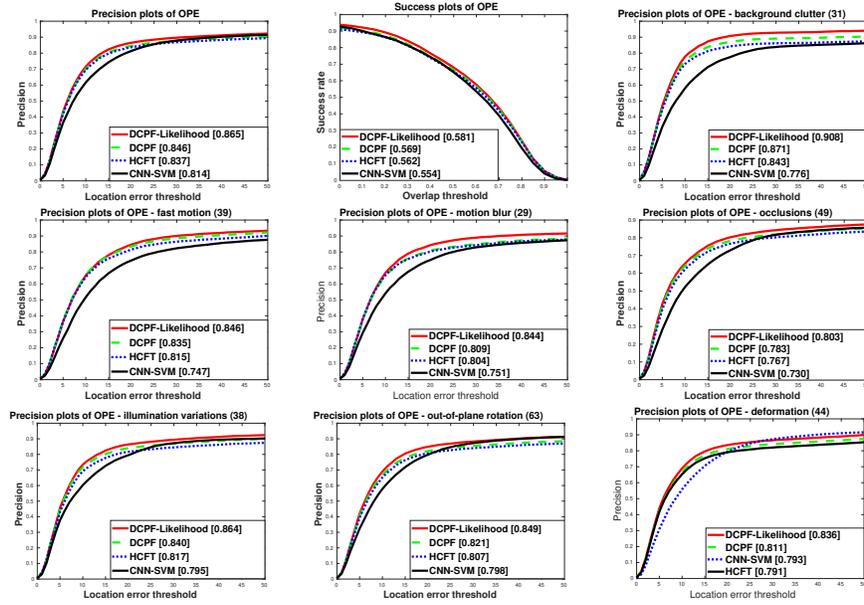


Fig. 7. One pass evaluation of our tracker in comparison with three state-of-the-art approaches.

qualitative results comparing our tracker with DCPF [8], HCFT [7], and CNN-SVM [4]. In both data sequences shown in the figure, our method successfully handles occlusion scenarios. These results highlight the impact of using more reliable sampling distributions.

Fig. 7 shows the OPE results for our tracker in comparison with DCPF, HCFT, and CNN-SVM. Our overall performance improvements over DCPF, the second best tracker, in terms of precision and success rates are 2.5% and 2%, respectively. Our method outperforms DCPF particularly in scenarios involving occlusions (+3%) and background clutter (+4.5%). DCPF uses the transition distribution as the proposal density, a common approach in particle-correlation trackers. Our results show that the likelihood is a more effective proposal distribution. In scenarios involving motion blur and fast motion, our performance improvements over DCPF are around 4.5% and 2%, respectively, because our tracker increases the variance of the likelihood distribution to spread out particles across a wider area. Our method also outperforms DCPF in scenarios involving illumination variation (+3%), out-of-plane rotation (+3.5%), and deformation (+3%). Our method also decreases the computational cost of the algorithm. Our tracker uses 100 particles, which is significantly less than the 300 particles used in DCPF.

5 Conclusion

In this work, we propose the DCPF-Likelihood visual tracker. Our method estimates a likelihood distribution as the proposal density for a particle filter based on correlation

response maps. Correlation response maps provide an initial estimate of the target location, which results in more accurate particles. Furthermore, the resulting likelihood distribution has a wider variance in challenging scenarios such as fast motion and motion blur. Our particle filter also generates a likelihood distribution for each correlation map cluster in difficult scenarios such as target occlusions. Our results on the OTB100 dataset show that our proposed visual tracker outperforms state-of-the-art methods.

References

1. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing* **50**(2), 174–188 (2002)
2. Dai, K., Wang, D., Lu, H., Sun, C., Li, J.: Visual tracking via adaptive spatially-regularized correlation filters. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4670–4679 (2019)
3. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **37**(3), 583–596 (2015)
4. Hong, S., You, T., Kwak, S., Han, B.: Online tracking by learning discriminative saliency map with convolutional neural network. In: *32nd International Conference on Machine Learning* (2015)
5. Kawabata, T.: Multiple subunit fitting into a low-resolution density map of a macromolecular complex using a gaussian mixture model. *Biophysical Journal* **95**(10), 4643–4658 (2008)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (eds.) *Advances in Neural Information Processing Systems 25*, pp. 1097–1105 (2012)
7. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: *IEEE International Conference on Computer Vision (ICCV)* (2015)
8. Mozhdehi, R.J., Medeiros, H.: Deep convolutional particle filter for visual tracking. In: *24th IEEE International Conference on Image Processing (ICIP)* (2017)
9. Mozhdehi, R.J., Reznichenko, Y., Siddique, A., Medeiros, H.: Convolutional adaptive particle filter with multiple models for visual tracking. In: *13th International Symposium on Visual Computing (ISVC)* (2018)
10. Mozhdehi, R.J., Reznichenko, Y., Siddique, A., Medeiros, H.: Deep convolutional particle filter with adaptive correlation maps for visual tracking. In: *25th IEEE International Conference on Image Processing (ICIP)* (2018)
11. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., Yang, M.H.: Hedged deep tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4303–4311 (2016). DOI 10.1109/CVPR.2016.466
12. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: *International Conference on Learning Representations (ICLR)* (2015)
13. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2013)
14. Yuan, D., Lu, X., Liang, Y., Zhang, X.: Particle filter re-detection for visual tracking via correlation filters. *Multimedia Tools and Applications* (2018)
15. Zhang, M., Wang, Q., Xing, J., Gao, J., Peng, P., Hu, W., Maybank, S.: Visual tracking via spatially aligned correlation filters network. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 469–485 (2018)
16. Zhang, T., Liu, S., Xu, C.: Correlation particle filter for visual tracking. *IEEE Transactions on Image Processing* **27**(6), 2676–2687 (2018)