Optimum Correlation filters for Visual Tracking via Histogram of Gradient Features and SVM classifier

1st Djamel Eddine Touil
Energy Systems Modeling Laboratory
University of Biskra
Biskra, ALGERIA
tde.touil@gmail.com

2nd Nadjiba TERKI
Department of electrical engineering
University of Biskra
Biskra, ALGERIA
t_nadjiba@yahoo.fr

3rd Riadh AJGOU
Department of Electrical engineering
University of El-Oued
El-Oued, ALGERIA
riadh-ajgou@univ-eloued.dz

4th Khelil SIDI BRAHIM
Energy Systems Modeling Laboratory
University of Biskra
Biskra, ALGERIA
khelilsidibrahim@hotmail.com

***Abstract*—The majority of object trackers algorithms cannot recover tracking processes from problems of drifting. These problems are caused by several challenges, especially heavy occlusion, scale variation, and fast motion. In this paper, we present a new effective method with the aim of treating these challenges robustly basing on two principal tasks. First, we infer the target location using the correlation map, resulting from the combination of a learned correlation filter model with a histogram of gradient (HOG) features. Indeed, Bat algorithm (BA) is exploited for solving the update model equation of the correlation filters. Second, we use the histogram of gradient features to learn another correlation filter model in order to estimate the scale variation. Furthermore, we exploit an online training SVM classifier to re-detect target in failure cases. The extensive experiments on a commonly used tracking benchmark dataset justify that our tracker significantly outperforms the state-of-the-art trackers.**

***Index Terms*—HOG, Correlation filter, BA, SVM, Visual tracking.**

I. Introduction

It is commonly known that computer vision is a wide field of active applications; recently, visual tracking is playing the important role including research topics, such as video processing, motion analysis, and unmanned control systems [1]. The performance of visual trackers, either in single or in multiple objects tracking classes, is being significantly ameliorated including robustness, accurateness, and execution speed. Some challenges, however, such as heavy occlusions, non-rigid deformations, fast motion, background clutters, etc, are still cause strict weakness for many tracking methods. Since the category of the tracked object is unknown, the most common paradigm is to learn an appearance behavior of the unknown tracked object in an online mode using features extracted from the sequence itself [2]. In general, visual tracking methods are categorized as a generative [3]–[5] and discriminative approach [6]–[9]. Discriminative methods have witnessed an important development basing on correlation filters [10], [11]. The target is determined as the area that has the high response when a correlation occurs between target features and correlation filter models. In [12], they improved clearly the tracking performance in different challenging cases, but they fail in the presence of scale variations. Thus various works tried to handle accurately this issue, however, the tracking speed was reduced [13], [14]. Recently, correlation filters have been widely used [15]. In visual tracking, correlation filters have attracted considerable attention due to its computational efficiency. Bolme et al. propose to learn a minimum output sum of squared error (MOSSE) [11] filter for visual tracking, applied to gray-scale images, where the learned filter encodes target appearance with its update on every frame. Through the use of correlation filters, the MOSSE tracker is computationally efficient in speed reaching, several hundred frames per second. The CSK method builds on illumination intensity features and is further improved by using HOG features in the KCF tracking algorithm [12]. Recently, Zhang et al. [16] incorporate the information context in filter learning and the scale change is modeled by consecutive correlation responses. The DSST tracker [13] learns adaptive multi-scale correlation filters using HOG features to handle the scale change of target objects. However, these methods do not address the critical issues regarding an online model update. They are susceptible to be less effective for handling long-term occlusion, unable to handle the drifting and out-of-view problems. We propose a novel-tracking algorithm for handling these challenges cases. The following is a summary of our main contributions: First, we propose to improve the detection module by learning an SVM classifier to re-detect the target in tracking failure cases, specifically for heavy occlusion. Second, in order to handle the target's scale variation, we use the Histogram of gradient [17] as features to learn adaptive multi-scale correlation filters to estimate an optimal scale. Third, we propose to employ the BAT algorithm (BA) [18], [19] in optimizing a fitness function computed between two consecutive frames. The objective is to learn correlation filters in the spatial domain. We show that it is effective in handling challenges of background clutters, fast motion deformation. Fourth, we evaluate our tracking algorithm on two reference datasets OTB-50 [20] and VOT 2016 [21].

The results of the experiments justify that our method works in a favorable way against many state-of-the-art methods. The rest of this paper is organized as follows. In Section 2, we present in detail our proposed method. Section 3 discusses the use of the online re-detector. In the fourth Section, we detail the use of bat algorithm. We demonstrate the effectiveness of our tracker, in Section 5, by giving a clear comparison of results with several reference trackers on two reference datasets. Finally, Section 6 gives the conclusion and future work.

II. Proposed Algorithm

In this section, we present our method, which is capable to handle several challenging appearance changes of the target, especially; the heavy occlusion, fast motion, and scale variation. The core of our tracking algorithm can be summarized in three essential steps. First, unlike in [17], we estimate the target location by learning two-dimensional correlation filter with HOGs features. We propose the Bat algorithm for learning correlation filter model. Second, we estimate the appropriate scale upon one-dimensional correlation filters with the histogram of gradients (HOG) features. Figure 1 clarifies the details of our tracking method.

A. Correlation filters

Due to its improvement of precision and running speed, correlation filter has known a significant interest in object tracking task [22]. Furthermore, the state-of-the-art object trackers that achieve the highest results exploit correlation filter [23]. The objective is encoding the target appearance by learning a correlation filter from the previously obtained features of targets and their background, then hint the targets location by searching the location of the peak of the filter response map [11], [24]. For this aim, we adopt filter w trained on an image patch x of size $M \times N$ Each training samples are derived from all the circular shifts $x_{m,n}(m,n) = \{0,1,...,M-1\} \times \{0,1,...,N-1\}$. All circular shifts are labeled $y_{m,n}$ using a Gaussian function [25], where $y(m,n) = e^{-\frac{(m-\frac{M}{2})^2+(n-\frac{N}{2})^2}{2\sigma^2}}$. The solution of the following minimization problem is used to learn a correlation filter w with the same size of x.

$$w^* = \arg\min_w \sum_{m,n} |w.x_{m,n} - y(m,n)|^2 + \lambda|w|^2 \quad (1)$$

where λ is a regularization parameter ($\lambda > 0$). The fast Fourier transformation (FFT) is adopted in Eq.(1) to exploit the fact that the auto-correlation of a signal x in Fourier domain represents the power spectrum [12], [25].

$$W^d = \frac{Y \bigodot \overline{X}^d}{\sum_{i=1}^{D} X^i \bigodot \overline{X}^i + \lambda} \quad (2)$$

Where each capital letter refers to the corresponding Fourier transformed signal, the bar denotes the complex conjugation and the operator $\bigodot$ is the Hadamard product. For each new frame, the tracking process is performed on multichannel feature vector Z^l on the l^{th} layer computed the l^{th} correlation response, i.e.,

$$f_l = F^{-1}(\sum_{d=1}^{D} W^d \bigodot \overline{Z}^d) \quad (3)$$

The operator F^{-1} denotes the inverse FFT. Therefore, the new target location is detected basing on where the peak of the response map f of size $M \times N$ appears.

B. Histogram of gradient

In our work, we adopt the histogram of gradient (HOG) [17] for the optimal scale estimation. The procedure of extraction HOG is as follows.

- To improve the contrast of images, we use Gamma transformation. It does a conversion of the gray level of images. The transformation function is:

$$y(n_1,n_2) = x^{\epsilon}(n_1,n_2) \quad (4)$$

where $x(n_1,n_2)$ is the input image, and $y(n_1,n_2)$ is the output image. Here, we set the parameter $\epsilon = 0.5$

- The gradient can be computed by the following equations:

$$y_i(n_1,n_2) = h_i(n_1,n_2) \times x(n_1,n_2), i = 1,2 \quad (5)$$

where $h_i(n_1,n_2)$ is the derivative mask.

$$h_1(n_1,n_2) = [-1 \quad 0 \quad 1]^T, h_1(n_1,n_2) = [-1 \quad 0 \quad 1] \quad (6)$$

where h_1 and h_2 denote the discrete derivative masks in both the vertical and the horizontal directions. The overall gradient and orientation image are computed as:

$$y_m(n_1,n_2) = \sqrt{y_1^2(n_1,n_2) + y_2^2(n_1,n_2)} \quad (7)$$

$$y_\theta(n_1,n_2) = \arctan\left(y_1(n_1,n_2)/y_2(n_1,n_2)\right) \quad (8)$$

- The target zone in image input are partitioned into rectangular cells. Each cell is associated with an edge orientation histogram. For a cell histogram, the bin τ is calculated as:

$$H(\tau) = \sum_{i=1}^{N} y_m(n_1,n_2)\delta.[y_\theta^{'}(n_1,n_2) - \tau] \quad (9)$$

where δ is the Kronecker delta function and $y_\theta^{'}(n_1,n_2)$ is quantized orientation, computed from $y_\theta(n_1,n_2)$ and N is the total number of pixels in each cell [17]. Then the set of sums of magnitude in gradient τ is represented as an N-orientation histogram:

$$H^{all} = \{H(1), H(2), ..., H(N)\} \quad (10)$$

According to the $L2$-norm, the gradient values of each cell can be locally normalized in order to cope the illumination and contrast changes:

$$H_{hog}^{'}(\tau) = H_{hog}(\tau)/\sqrt{||H_{hog}(\tau)||_2 + \epsilon^2} \quad (11)$$

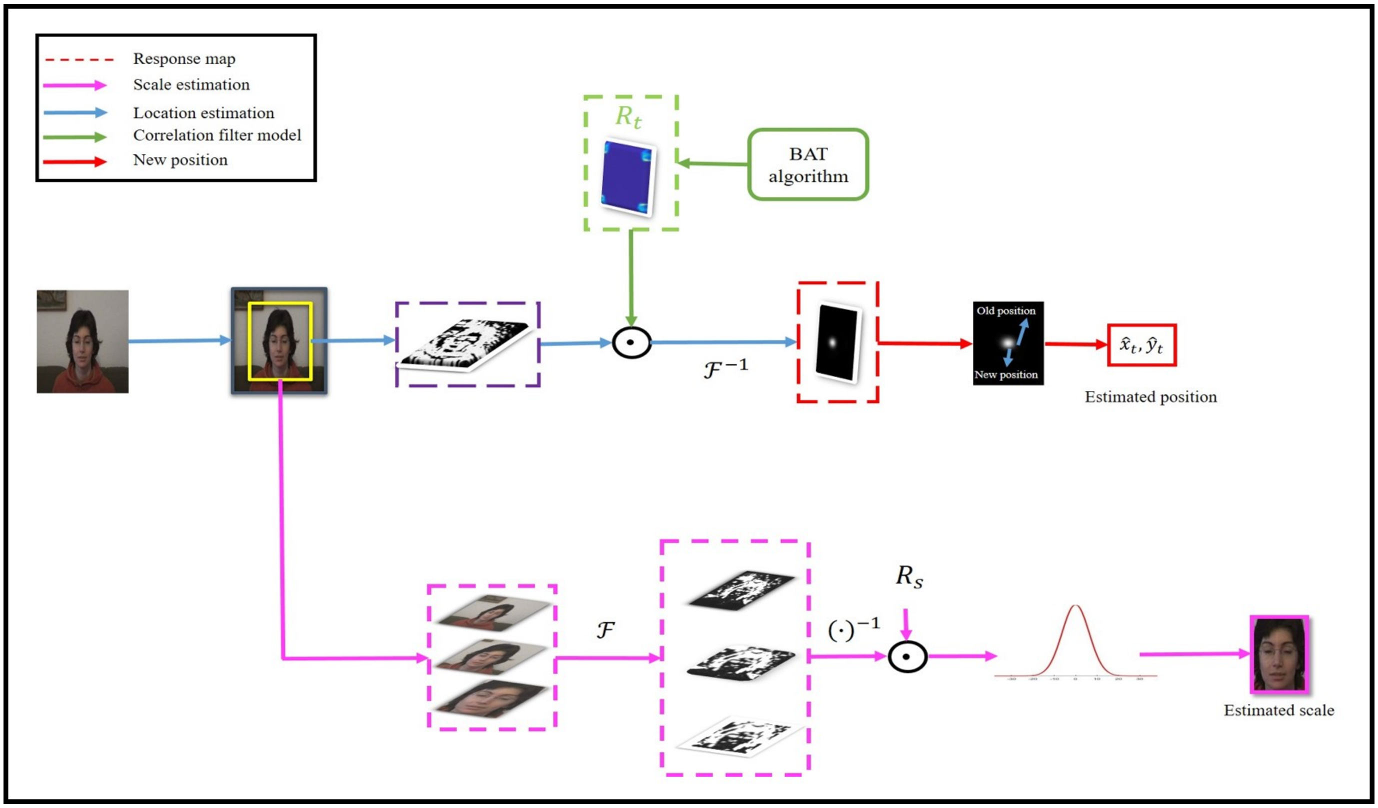

Fig. 1. Flowchart of the proposed method. The tracking task is decomposed into translation and scale estimation: translation is estimated using a translation model R_t, and scale is estimated by the scale model R_s. For modeling the correlation filters efficiency,correlation filters model R_t in green rectangle is optimized through Bat algorithm.The response map is used to estimate the target location by finding the position of its maximum value.

In our implementation, we concatenate HOG in a 31 bins [26]. In addition, another HOG features are concatenated in 16 bins in order to provide accurateness to severe illumination changes. Therefore, vectors with 63 channels are used to train the scale model R_s.

C. Target location estimation

We learn the correlation filter model R_t with feature map to obtain the response maps basing on (3). Then, the target estimated position $(\widehat{x_p}, \widehat{y_p})$ is inferred by searching the maximal value of Eq.3 to determine the confidence of tracking results.

D. Target Scale Estimation

For fast tracking of the target scale variation, we exploit HOG features [17] to train Three-dimensional scale space correlation filter model R_s of fixed size $P \times Q \times S$, where P and Q are the height and width of the model and S is the number of scales. The scale estimation task is detailed as follows. First, we build a target pyramid around the new estimated targets position using HOG features. Let us consider that is the scale factor and K represent the number of candidate scales S, where $S = \{a^n | n = -\frac{K-1}{2}, -\frac{K-3}{2}, ..., \frac{K-1}{2}\}$. For each $s \in S$, an image patch J_s of size $sP \times sQ$ is extracted and centered on the estimated position $(\widehat{x_p}, \widehat{y_p})$ [27]. Then, all patches are resized with size $P \times Q$ again. Second, we train the model Rs in the frequency domain using fast Fourier transform (FFT) using Eq.(2). Third, we weight R_s by a Three-dimensional Gaussian function to construct the desired correlation response map $\widehat{f}_s$ of the scale model R_s. The optimal scale $\widehat{S}$ of target is derived using (12). Finally, the scale model R_s is updated using Eq.(13).

$$\widehat{s} = \arg\max_{s}(max(\widehat{f_1}), max(\widehat{f_2}), ..., max(\widehat{f_K}) \tag{12}$$

E. Models update

To maintain long-term tracking, it is inevitable for both translation model R_t and scale model R_s to be adaptive in the two estimation steps [25]. We update R_t and then R_s frame by frame with a learning rate η as in Eq.(13). A predefined threshold T_a is used to update only R_s if $max(f) > T_a$.

$$W_t^d = (1-\eta)W_{t-1}^d + \eta \frac{Y \odot \overline{X}^d}{\sum_{i=1}^{D} X^i \odot \overline{X}^i + \lambda} \tag{13}$$

III. Online Re-detector

A. Samples collection

We propose two main classes of samples to re-detect tracked object by a given feature points extracted from the image patches. We use the SIFT vector [28], computed from local HOG to extract these samples. In fact, it works well on textured images. Similar to [29], we use key-points as samples for image retrieval in very large image databases. Key-points are labeled by a SVM classifier [9] based on their SIFT descriptors. Then we localize individually the key-points in order to obtain the location information which is necessary to re-detect the tracked object.

B. Classification formulation

As presented in subsection (3.1), we use the extracted key-points to classify all appearances of the image patch surrounding these key-points as positive samples. Then, we treat the next given patch surrounding the detected key-points in the actual frame to refer it to the most likely class. We adopt an online random feature classifier in order to train the detector on the entire frame using sliding windows strategy when $max(\widehat{f_s}) < T_r$ [25]. Let denote $c_i, i \in \{-1, 1\}$ as the indicator of class labels, each input sample z is represented by a binary feature vector $f_{ej}(z)$. Formally, we look for finding positive samples z^* which are sorted using an SVM classifier [9] if their class c is the solution of Eq.(14) with N selected features.

$$\widehat{c} = argmax(g(f_{ej}(z))) \tag{14}$$

where, g is a linear function $g(f_{ej}(z)) =< w, \phi(z) >$, in which w is the SVMs weight vector and is kernel map which maps z into another suitable feature space z^* and is implicitly defined by a joint kernel function [9].

$$K(z, z^*) =< \phi(z), \phi^*(z) > \tag{15}$$

IV. BAT ALGORITHM (BA)

The Bat algorithm (BA) is a new meta-heuristic algorithm. It simulates the echolocation characteristics of bats with different emission rates and pulse intensity [18]. Since its first implementation, BA has proven its superiority in performing and solving, compared to other well-known algorithms, a wide range of global optimization issues. In addition, studies show that this algorithm has become an active search axis due to the appearance of several BA extensions. In this section, we focus on the Bat algorithms details used in our work. The main idea is the use of BA behavior to optimize (1) in order to achieve the appropriate correlation filter models.

A. Procedure

The basic of BAT algorithm is summarized in three rules [18].

- All bats use echolocation to detect distance, and they can also "know" the difference between food/prey and obstacles;
- For prey searching, bats fly randomly with velocity v_i at position x_i with a fixed frequency and loudness. According to the proximity of the target, they adjust automatically the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission.
- It is assumed that the loudness varies from a large (positive) A_0 to a minimum constant value A_{min}.

The position x^{t-1} and velocity v^{t-1} of each bat are defined in a d-dimensional search space, and they are updated subsequently during the iterations. The new solutions x_i^t, M_i^t and velocities v_i^t at time step t are calculated using a global search strategy by [19]:

Algorithm 1 Pseudo-code of bat algorithm

Input: $Problem_{size}$, HOG's features, correlation filter
Output: W^*
```
    for i = 1 to n do
        x_i ← rand(Problem_size, n);
        v_i ← rand(Problem_size, n);
        Define f_i at x_i;
        Initialize the pulse rates r_i^0 and the loudness A_i^0;
        Generate W_i(n, Problem_size, HOG'sfeatures);
    end
while (iteration number < 200)
    Generate new solution according to Eq.s (16-19);
        if rand > r_i do
        Select a solution among the best solutions;
        Generate a local solution around the selected;
        best solution according to Eq.(20);
    end
    Generate new solution by flying randomly;
    if (rand > A)&(f(x_i) < f(x*))&(f(W_i) < f(W*)) do
        accept the new solutions;
        Generate a local solution around the selected;
        increase r_i and decrease A_i using Eq.(21);
    end
    Extract the current best x*, W* from the best bat;
end
return W*;
```

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{16}$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x^*)f_i \tag{17}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{18}$$

$$W_i^t = W_i^{t-1} + v_i^t \tag{19}$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. x^* is the current global best solution among all then bats at the current iteration. f_i is the frequency value of the i^{th} bat and W_i^t is a correlation filter model generated for each bat in time step t. For the local search part, once a solution is selected from among the current best solutions, a new solution for each bat is generated locally using a random walk as in Eq.(20).

$$x_{new} = x_{old} + \varepsilon A^t$$
$$W_{new} = W_{old} + \varepsilon A^t \tag{20}$$

where $\varepsilon \in [0, 1]$ is a random number, x_{old}, W_{old} are the solutions in the current optimization solution set, and A^t is the average loudness of all the bats at time step t. In addition, loudness A_i and the pulse rate r_i are updated if the target is found, i.e.,

$$A_i^{t+1} = \omega A_i^t$$
$$r_i^{t+1} = r_i^0[1 - e^{-\lambda t}] \tag{21}$$

where r_i^0 is the initial pulse rate, ω is the pulse frequency increasing coefficient and λ is the pulse amplitude attenuation coefficient. For any $\omega > 0$, and $\lambda < 1$ and for $t \to \infty$, we have:

$$A_i^t \to 0 \quad \text{and} \quad r_i^t \to r_i^0 \tag{22}$$

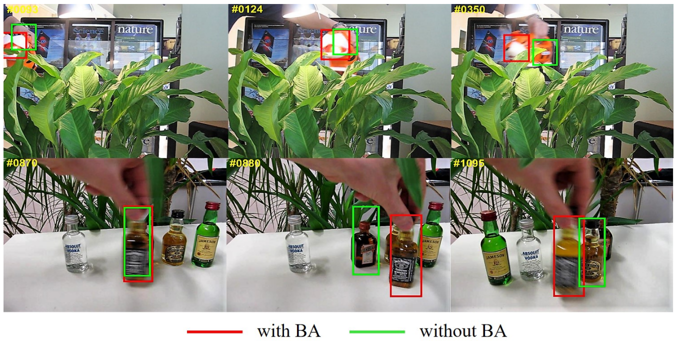

Fig. 2. Comparisons of our proposed approach using and without using BAT algorithm in challenging situations. It is clear that our tracker performs efficiently in handling background clutter and deformation in 93^{th}, 124 and 350 frames respectively in *Tiger 2* sequence. Likewise, it performs robust to treat occlusion and fast motion in 870^{th}, 880^{th} and 1095^{th} frames respectively on *Liquor* sequence.

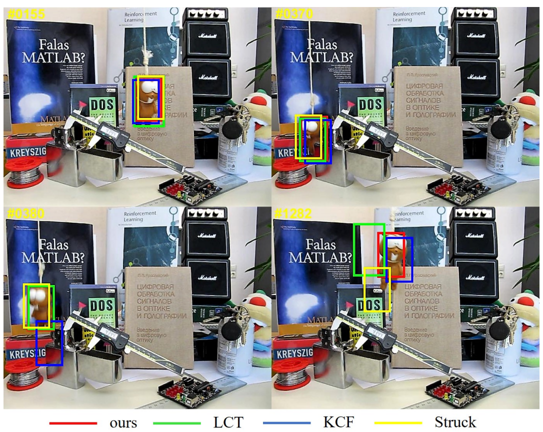

Fig. 3. Example of handling challenging cases of fast motion and heavy-occlusion with three state-of-the-art trackers LCT [25], KCF [12] and struck [9] on the *Lemming* sequence [20]. It is clear that our tracker outperforms the other tracker.

Based on these idealized rules, the basic steps of BA can be summarized as the pseudo-scheme code shown in Algorithm 1.

B. Correlation filter model

As shown in Algorithm 1, each bat is initialized with a set of candidate model, with a size adapted to the HOGs feature map. We use 10 candidate models for each particle. Noting that this initialization is done using the previous correlation filter model in the spatial domain. Once the model is initialized, we evaluate Eq.(1) in order to choose the best model. By resorting to 200 iterations, the BAT algorithms dynamic can effectively meet the suitable correlation filter model, which gives an accurate locating of the target in most of all frames. In fact, this behavior could lead to a more efficient optimization of the correlation filter model as the solution of (3) in which Fourier domain did not. For example, Figure 2 shows that our BA-based method is effective to handle object appearance changes, especially; background clutters, fast motion, deformation, and occlusion.

TABLE I
BA PARAMETERS

Parameter	Value
Population size	25
Pulsation rate r	0.55
Loudness A	0.25
Alpha α	0.9
Gama γ	0.99

V. EXPERIMENTAL RESULTS

The regularization parameter λ in Eq.(1) is set to10^{-4}. The learning rate η in Eq.(13) is set to 0.01. The number of scale space is $|s|$ and the scale factor a is set to 1.049. To activate the trained feature detector, we use the threshold $T_r = 0.2$. A threshold $T_t 0.5$ is used to adopt the re-detection results. We fix $T_a = 0.45$ for update the scale model in Eq.(13). BAT algorithms parameters are summarized in Table 1. We implement our tracker in Matlab with Intel (R) Core (TM) i7-6700K 4.00GHz CPU with 32GB RAM.

A. Experiment on OTB-50

We evaluate experiments results using distance precision (DP) and overlap success (OS) metrics [20]. DP presents the percentage of frames whose estimated location at a threshold of 20. OP is defined as the percentage of frames where the bounding box overlaps exceeds a given threshold $t \in [0, 1]$. OPE results are presented at $t = 0.5$. We fairly compare it with 8 reference methods, including TLD [7], MEEM [8], Struck [9], KCF [12], LCT [25], TGPR [30], SCM [31] and DLT [32]. The results are reported in one-pass-evolution (OPE) metric. It is obvious that our tracker outperforms the state-of-the-art methods. Figure 4 shows the comparison of OPE results on the OTB-50 dataset. The LCT tracker [25] achieves the second best results with an average distance precision of 84.8% and in overlap precision with 81.3%. Despite, the proposed method proves its efficiency with an amelioration gain of 1.1% in the average distance precision and with 0.3% in the average overlap precision. Further, Figure 3 shows that the insertion of a re-detection module by learning an SVM classifier allows the accurate re-detection of the target in long-term tracking case. We report results for four challenging tracking attributes in Figure 5. Among existing methods, The proposed tracker performs the best against the cited trackers in handling deformation, fast motion, scale variation and occlusion challenging cases.

B. Experiment on VOT 2016

We show in Figure 6 the comparison results in terms of expected average overlap (EAO) in VOT 2016 [21] of baseline trackers. For more details, please check [21]. It is clear that our proposed tracker exceeds the top tracker EBT [33] by an important gain. Table 2 lists the detailed results of our method and the top ranked state-of-the-art methods in VOT

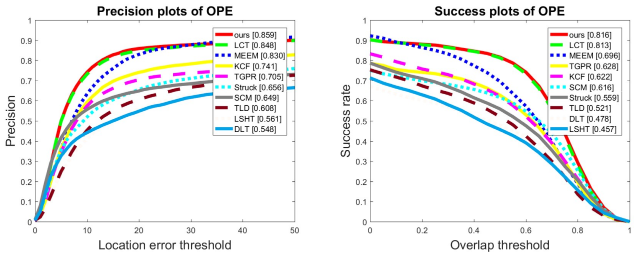

Fig. 4. Illustration of Distance precision and overlap success over the OTB-50 [20] dataset using OPE for each tracker.

2016: EBT [33], SRBT [34] and DNT [35]. The proposed tracker demonstrates its efficiency by achieving the best EAO score of 0.31. In addition, it provides a considerable percentage in overlap metric and the second best result in failures metric with 17.73%.

TABLE II
COMPARISON RESULTS IN TERMS OF EAO, OVERLAP AND FAILURES, WITH THE TOP REFERENCE TRACKERS ON THE VOT 2016 DATASET

Tracker	Overlap	Failures	EAO
Ours	0.53	17.73	0.31
EBT	0.45	15.19	0.29
SRBT	0.48	21.32	0.29
DNT	0.51	19.54	0.28

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel method for efficient object tracking. We propose a learned correlation filter model conservatively updated using HOG features. Moreover, we exploit a robust detector in order to recover the target from failure cases. Extensive experimental results show that our tracker outperforms the state-of-the-art trackers. Our future research will emphasis on making our method applicable in real time systems by using GPU and parallel implementation.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.

[2] D. Wu, W. Zou, X. Li, and Y. Zhao, "Kernalised multi-resolution convnet for visual tracking," *arXiv preprint arXiv:1708.00577*, 2017.

[3] X. Mei and H. Ling, "Robust visual tracking using 1 minimization," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1436–1443.

[4] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International journal of computer vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[5] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*. IEEE, 2012, pp. 2042–2049.

[6] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[7] Z. Kalal, K. Mikolajczyk, J. Matas *et al.*, "Tracking-learning-detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, p. 1409, 2012.

[8] J. Zhang, S. Ma, and S. Sclaroff, "Meem: robust tracking via multiple experts using entropy minimization," in *European Conference on Computer Vision*. Springer, 2014, pp. 188–203.

[9] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr, "Struck: Structured output tracking with kernels," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2096–2109, 2016.

[10] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *European conference on computer vision*. Springer, 2008, pp. 234–247.

[11] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2544–2550.

[12] 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.

[13] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.

[14] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *European conference on computer vision*. Springer, 2014, pp. 254–265.

[15] B. V. V. Kumar, M. Savvides, and C. Xie, "Correlation pattern recognition for face recognition," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1963–1976, 2006.

[16] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *European Conference on Computer Vision*. Springer, 2014, pp. 127–141.

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[18] X.-S. Yang and X. He, "Bat algorithm: literature review and applications," *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, pp. 141–149, 2013.

[19] X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.

[20] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.

[21] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin Zajc, T. Vojir, G. Häger, A. Lukežič, and G. Fernandez, "The visual object tracking vot2016 challenge results," Springer, Oct 2016. [Online]. Available: http://www.springer.com/gp/book/9783319488806

[22] W. Huang, J. Gu, X. Ma, and Y. Li, "Correlation filter-based self-paced object tracking," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4437–4442.

[23] S. A. Wibowo, H. Lee, E. K. Kim, and S. Kim, "Multi-scale color features based on correlation filter for visual tracking," in *Signals and Systems (ICSigSys), 2017 International Conference on*. IEEE, 2017, pp. 272–277.

[24] E. Gundogdu, H. Ozkan, and A. A. Alatan, "Extending correlation filter-based visual tracking by tree-structured ensemble and spatial windowing," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5270–5283, 2017.

[25] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5388–5396.

[26] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[27] Y. Li, Y. Zhang, Y. Xu, J. Wang, and Z. Miao, "Robust scale adaptive kernel correlation filter tracker with hierarchical convolutional features," *IEEE Signal Processing Letters*, vol. 23, no. 8, pp. 1136–1140, 2016.

[28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[29] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, vol. 2. Ieee, 2006, pp. 2161–2168.

[30] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *European Conference on Computer Vision*. Springer, 2014, pp. 188–203.

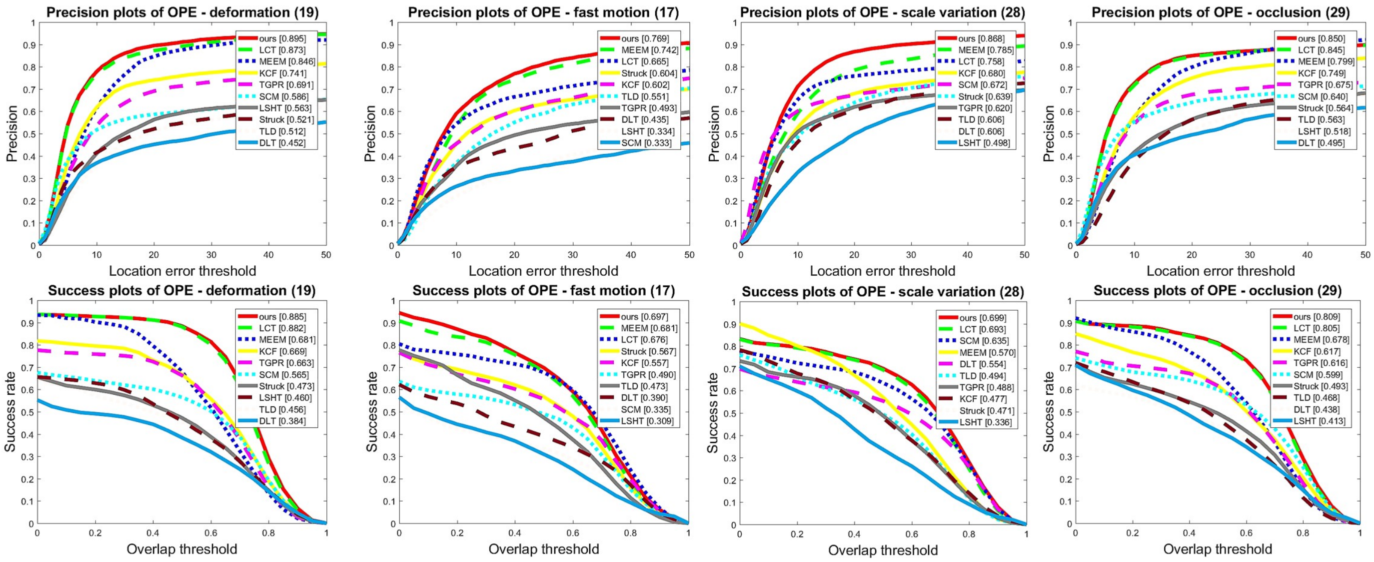

Fig. 5. Illustration of Distance precision and overlap success over four tracking challenges: deformation, fast motion, scale variation and occlusion. It is clear that our proposed method outperform the state-of-the art trackers.

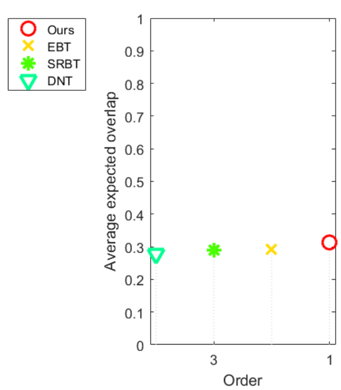

Fig. 6. Expected Average Overlap (EAO) graph with trackers ranked from right to left evaluated on VOT 2016.

[31] 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.

[32] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in neural information processing systems*, 2013, pp. 809–817.

[33] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 943–951.

[34] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder, "The visual object tracking vot2015 challenge results," in *Proceedings of the IEEE international conference on computer vision workshops*, 2015, pp. 1–23.

[35] Z. Chi, H. Li, H. Lu, and M.-H. Yang, "Dual deep network for visual tracking." *IEEE Trans. Image Processing*, vol. 26, no. 4, pp. 2005–2015, 2017.