2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

OWP: Objectness Weighted Patch Descriptor for Visual Tracking

Bo Jiang
School of Computer Science and Technology
Anhui University, China
jiangbo@ahu.edu.cn

Yuan Zhang
School of Computer Science and Technology
Anhui University, China
zhangyuanahu@163.com

Jin Tang
School of Computer Science and Technology
Anhui University, China
tj@ahu.edu.cn

Bin Luo
School of Computer Science and Technology
Anhui University, China
luobin@ahu.edu.cn

***Abstract*—Visual object tracking is an active research problem and has been widely used in computer vision and pattern recognition area. Existing visual tracking methods usually localize the visual object with a bounding box which are often disturbed by the introduced background information and partial occlusion because of bounding box representation of visual object. To deal with this problem, in this paper, we propose a novel Objectness Weighted Patch (OWP) descriptor for object feature descriptor in visual tracking. The aim of OWP is to assign different objectness weights to the patches of bounding box to reduce the influences of background information and partial occlusion. We propose to compute the objectness weights of patches in OWP by integrating multiple cues (background, foreground and local spatial consistency) together in a general optimization model. Also, the proposed model has a simple closed-form solution and thus can be computed efficiently. We incorporate our OWP into structured SVM tracking framework and provide a new robust tracking method. Extensive experiments on two standard benchmark datasets OTB100 and Temple-Color demonstrate the effectiveness and benefits of the proposed tracking method.**

I. Introduction

Visual object tracking is an active research problem and has been widely used in computer vision and pattern recognition area. Existing tracking methods generally focus on the tracking-by-detection framework which aims to detect the target object with a bounding box from its background by using a classifier. In particular, the classifier is trained in the first frame using the ground truth bounding box, and updated in the subsequent frames by using the current tracking results. However, one important issue for this tracking-by-detection tracking framework is that it is usually sensitive to partial occlusion and background clutter. This is because (1) the bounding box is general inaccurately to describe the target object due to the irregular shape of target object and thus contains some background information. (2) There may also exist some partial occlusions in bounding box [1].

To overcome this issue, one kind of popular methods is to alleviate the disturbance of partial occlusion or unavoidable background information in bounding box feature extraction [2], [3], [4], [5], [6], [7], [8], [1], [9]. For example, the methods proposed in [4], [5], [6], [7] update the object classifier by considering the distances of samples with respect to the bounding box center and assign higher weights to the samples that are close to the center. He et al. [2] present a key patches selection model according to location and occlusion to reduce the disturbance of partial occlusion and background information. Zhang et al. [3] propose to detect outlier patches in tracking problem by considering the movement information of local patches relative to the center. The methods proposed in [10], [11], [12] aim to segment the object from background to exclude the undesired effect of background information. Since the segmentation of object can be distributed by cluttered background, these methods are limited in dealing with cluttered background. Recently, Kim et al. [1] propose to construct a graph model to represent the patches of bounding box, and then use a random walk with restart (RWR) model to obtain different weights for the patches to alleviate the influences of background effects and partial occlusion. However, one limitation of RWR is that it generally fails to consider the local consistency of patches with similar appearance in its weight computation. Li et al. [13] further propose to learn a low-rank sparse graph for target object representation which results in more robust patch weight computation and tracking results. They recently also [14] propose an improvement model of graph learning by furthering considering both local and global cues in graph learning process. One limitation of these two methods is that they need an iteration algorithm to learn an optimal graph under the low-rank sparse constraint which has high computational complexity and thus reduces the efficiency of their tracking methods.

Inspired by recent works [1], [13], [14], in this paper, we propose a novel efficient objectness weighted patch descriptor (OWP) for visual tracking problem. The aim of OWP is to assign each patch of bounding box with an objectness weight to represent how likely it belongs to the target object and thus alleviate the undesired disturbance of partial occlusion and cluttered background information. To compute the objectness of patches, we propose to use a principled optimization model which integrates multiple cues (background, foreground and local consistency) together in objectness computation. Also, the proposed model has a closed-form solution and thus can be computed efficiently. Then, the computed objectness weights are combined with patch features to construct objectness weighted patch descriptor for visual tracking. Extensive experiments on two standard benchmark datasets show the effectiveness and efficiency of the proposed tracking method.

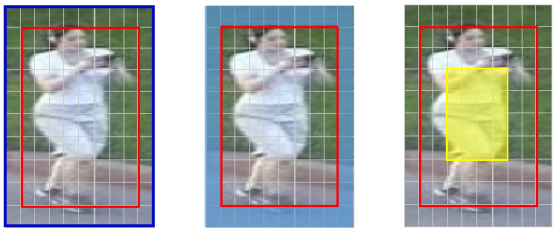
978-1-5386-3788-3/18/$31.00 ©2018 IEEE

II. Related works

Here, we only briefly review some related tracking methods which aim to alleviate the influence of cluttered background information in feature representation. Comaniciu et al. [4] propose to assign smaller weights to boundary pixels in its histogram feature extraction by using a kernel method. Similar way has also been used in work [6], which aims to assign smaller weights to the pixels that are far from the center of the bounding box. To reduce the influence of partial occlusion, He et al. [2] propose a key patch selection method by considering the location and occlusion. Zhang et al. [3] propose to detect and reduce the effect of outlier patches in tracking problem by exploring the relative movement information of local patches. Kim et al. [1] aim to assign foreground weights to the patches of bounding box by using a random walk with restart model to alleviate the influence of the background. Then, they generate a kind of weighted patch descriptor for visual tracking. Li et al. [13] propose to use a semi-supervised learning method to learn an optimal low-rank sparse graph for object representation. They recently also [14] propose an improvement graph learning model by further considering both local and global cues simultaneously.

III. Objectness Measurement of Patches

Given one bounding box c of the target object, we first partition it into n non-overlapping patches $\{p_1, p_2, \cdots p_n\}$, and then assign each patch p_i with an objectness weight $\mathbf{w}_i$ to represent its possibility of belonging to the target object. Then, we combine the computed objectness weights with patch features to construct objectness weighted patch descriptor (OWP) for visual tracking. In this section, we propose our optimization model to compute the objectness weights for the patches of bounding box. The proposed model aims to explore background cue, foreground cue and local consistent constraint simultaneously in objectness computation.



(a) (b) (c)

Figure 1: (a) Original bounding box c (red box) and expended bounding box c' (blue box); (b) Expended region $\mathrm{R}^{\mathrm{out}}$(blue region); (c) Shrunk region R^{in}(yellow region).

A. Background cue

Given one bounding box c, we first define an expended bounding box c', as shown in Figure 1(a). Similar to the observation in previous works [1][15], the patches near the bounding box boundary are likely to belong to background, i.e., many patches in expended region $\mathrm{R}^{\mathrm{out}}$ belong to background, as shown in Figure 1(b). Thus, we can define an initial background weight distribution $\tilde{\mathbf{u}}_i$ for patches of the expended bounding box c' as follows,

$$\tilde{\mathbf{u}}_i = \begin{cases} \frac{1}{n_{\mathrm{out}}} & \text{if patch } p_i \in \mathrm{R}^{\mathrm{out}} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where n_{out} denotes the number of patches in $\mathrm{R}^{\mathrm{out}}$. To obtain a kind of more reasonable background measurement for original bounding box c, we employ a graph diffusion technique to conduct weight diffusion based on the initial weight distribution $\tilde{\mathbf{u}}$. Many diffusion models [16], [15], [17] can be used here. Here, we utilize random walk with restart model [18], [1]. To do so, a neighborhood graph is constructed whose nodes represent patches of expended bounding box c' and edges denote the 8-neighbourhood relationship between patches. The weighted adjacency matrix $\mathbf{W}$ is computed as $\mathbf{W}_{ij} = \exp(-\eta \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2), \eta = 5$ where $\mathbf{x}_i$ and $\mathbf{x}_j$ denote the features of patch p_i and p_j, respectively. By normalizing each column of weight matrix $\mathbf{W}$ to 1, we can obtain transition matrix $\mathbf{A}$ as, $\mathbf{A}_{ij} = \mathbf{W}_{ij} / \sum_{i=1}^{n} \mathbf{W}_{ij}$. Using transition matrix $\mathbf{A}$ and initial distribution $\tilde{\mathbf{u}}$, we can conduct diffusion as,

$$\mathbf{u}^{(t+1)} \leftarrow \alpha \mathbf{A}\mathbf{u}^{(t)} + (1-\alpha)\tilde{\mathbf{u}} \tag{2}$$

where $\alpha \in [0, 1]$ is the restart probability. Note that, the converged diffusion $\mathbf{u}$ is given by

$$\mathbf{u} = (1-\alpha)(\mathbf{I} - \alpha\mathbf{A})^{-1}\tilde{\mathbf{u}}. \tag{3}$$

where $\mathbf{I}$ is identity matrix. Thus, based on $\mathbf{u}$, we can obtain a kind of background measurement for original bounding box c. Similar diffusion has also been used in work [1]. Differently, here we use a different restart distribution $\tilde{\mathbf{u}}$ which is more simple. Also, the converged $\mathbf{u}$ is regarded as the background measurement which will be further used to guide the computation of objectness in our final model, as shown in the following Section C in detail.

B. Foreground cue

In addition to background cue, we can also obtain a kind of foreground measurement for patches. In general, the patches near the bounding box center are likely to belong to foreground [1]. One can thus define a shrunk region R^{in} of bounding box c (shown in Figure 1(c)), and the patches in R^{in} are likely to belong to foreground. Similarly, we can define an initial foreground weight distribution $\tilde{\mathbf{v}}$ for patches of bounding box c as follows,

$$\tilde{\mathbf{v}}_i = \begin{cases} \frac{1}{n_{\mathrm{in}}} & \text{if patch } p_i \in \mathrm{R}^{\mathrm{in}} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where n_{in} denotes the number of patches in R^{in}. We utilize random walk with restart model and obtain the converged diffusion $\mathbf{v}$ as

$$\mathbf{v} = (1-\alpha)(\mathbf{I} - \alpha\mathbf{A})^{-1}\tilde{\mathbf{v}} \tag{5}$$

where $\mathbf{I}$ is an identity matrix. The optimal $\mathbf{v}$ provides a kind of foreground measurement for patches of bounding box.

C. Objectness optimization

After obtaining the background and foreground measurement $\mathbf{u}, \mathbf{v}$, we then employ these two cues together to optimize the final objectness weight $\mathbf{w}$. In order to do so, we first construct a graph $G(V, E)$ with nodes V representing patches and edges E denoting the 8-neighborhood relationship between patch p_i and p_j. The weight of edge e_{ij} is calculated as,

$$\mathbf{S}_{ij} = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}) \tag{6}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ denote the feature descriptors of patch p_i and p_j, respectively. σ is a scaling parameter.

Our objectness computation is based on the following three observations: (1) large background measurement $\mathbf{u}_i$ of the patch p_i encourages it to take a small objectness weight $\mathbf{w}_i$; (2) Large foreground prior $\mathbf{v}_i$ encourages patch p_i to take a large objectness weight $\mathbf{w}_i$; (3) If patch p_i and p_j have similar appearance and are also 8-neighbors, then their corresponding objectness weights $\mathbf{w}_i$ and $\mathbf{w}_j$ should be close. Inspired by recent work on image saliency detection area [19], [17], we combine these three cues together by optimizing the following energy function,

$$\min_{\mathbf{w}} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{S}_{ij}(\mathbf{w}_i - \mathbf{w}_j)^2 + \sum_{i=1}^{n}\mathbf{u}_i\mathbf{w}_i^2 + \sum_{i=1}^{n}\mathbf{v}_i(\mathbf{w}_i - 1)^2 \tag{7}$$

This problem has a closed-form solution and the optimal solution $\mathbf{w}^*$ can be obtained by setting the first derivative of the energy function w.r.t. variable $\mathbf{w}$ to $\mathbf{0}$, i.e.,

$$\mathbf{w}^* = \big(\mathbf{D} - \mathbf{S} + \mathbf{U} + \mathbf{V}\big)^{-1}\mathbf{v} \tag{8}$$

where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \sum_{j=1}^{n}\mathbf{S}_{ij}$. Matrix $\mathbf{U}$ and $\mathbf{V}$ are diagonal matrices with $\mathbf{U}_{ii} = \mathbf{u}_i$ and $\mathbf{V}_{ii} = \mathbf{v}_i$.

IV. OWP Descriptor and Tracking

We incorporate the computed patch objectness weights into the tracking-by-detection method and propose a robust tracking approach. In general, our tracking process contains three main steps, i.e., OWP descriptor, structured SVM tracking and scale estimation.

A. OWP descriptor

Given bounding box c, we first extract $\mathbf{X}_c = (\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_n)$ as the feature descriptor for it, where $\mathbf{x}_i$ denotes the descriptor of the i-th patch. Based on the objectness weights of patches, we can obtain a kind of robust weighted descriptor for bounding box c as

$$\mathbf{X}_c^w = (\mathbf{w}_1\mathbf{x}_1, \mathbf{w}_2\mathbf{x}_2, \cdots \mathbf{w}_n\mathbf{x}_n) \tag{9}$$

Comparing with original feature descriptor $\mathbf{X}_c$, in $\mathbf{X}_c^w$ the undesired background patches included in bounding box c are alleviated due to the small objectness weights.

B. Structured SVM tracking

The proposed OWP descriptor can be incorporated into many traditional tracking-by-detection methods. Here, we incorporate it into Struck [5] tracking method, which aims to adopt structured SVM method during classification. Formally, Struck [5] selects the optimal target bounding box c_t^* in the t-th frame by maximizing a classification score,

$$c_t^* = \text{argmax}_c\langle \mathbf{h}_{t-1}, \mathbf{X}_{t,c}^w\rangle \tag{10}$$

where $\mathbf{h}_{t-1}$ is the normal vector of a decision plane of $(t-1)$-th frame, and $\mathbf{X}_{t,c}^w$ denotes the OWP descriptor of bounding box c in the t-th frame. In order to further incorporate the information of the initial frame with ground truth bounding box, we instead compute the optimal bounding box c_t^* by using the following balancing score,

$$c_t^* = \text{argmax}_c\big(\epsilon\langle \mathbf{h}_{t-1}, \mathbf{X}_{t,c}^w\rangle + (1-\epsilon)\langle \mathbf{h}_0, \mathbf{X}_{t,c}^w\rangle\big), \tag{11}$$

where $\mathbf{h}_0$ is learnt in the initial frame which can prevent it from learning drastic appearance changes. We set $\epsilon = 0.67$ in this paper. After obtaining the optimal bounding box c_t^*, we then update the classifier $\mathbf{h}_t$. To prevent the effects of unreliable tracking results, we update the classifier only when the confidence score of tracking is larger than a threshold $\theta = 0.3$, as suggested in works [1], [13].

C. Scale estimation

To deal with scale variation problem, we conduct a scale estimation in our tracking process. Similar to the previous work [20], we first estimate the center of a target using a fixed scale, and then generate a set of candidate bounding boxes at different scales. Finally, we determine the optimal scale of the target at the estimated center location by using Struck [5] classifier. Despite the simplicity of this method, it can conduct scale estimation effectively and efficiently by avoiding large number of candidates.

V. Experimental Results

We implement all the experiments using C++ language on a desktop computer with an Inter i7 4.0GHz CPU and 32GB RAM. The proposed tracker performs at about 13.5 FPS (frames per second) which is faster than SOWP [1], DGT [13] and obviously faster than ReGLe [14]. Note that, our tracker performs faster than related work SOWP [1] because it uses a closed-form solution (Eqs.(3,5)) instead of iteration algorithm (Eq.(2)) used in SOWP to conduct diffusion of background and foreground measurements. We use the two commonly used benchmark datasets and protocols [25], [26] to evaluate the proposed tracker performance.

A. Evaluation settings

Parameters. For fair comparison, we fixed all parameters in all experiments. For each bounding box, we partition it into 8×8 (64) patches to obtain better result, as shown in Table II. For each patch, we extract 32-dimensional feature descriptor including 24-dimensional RGB color histogram and 8-dimensional oriented gradient histogram features. We scaled

Table I: Comparison results on attribute-based videos on OTB100 benchmark dataset. The attributes include IV (illumination variation), SV (scale variation), OCC (occlusion), DEF (deformation), MB(motion blur), FM (fast motion), IPR (in-plane-rotation), OPR (out-of-plane rotation), OV (out-of-view), BC (background clutters), and LR (low resolution). The red, green and blue colors indicate the best, second and third performances, respectively.

	Staple[21]	MEEM[8]	SOWP[1]	ReGLe[14]	ACFN[22]	LCT[20]	DLT[23]	HCF[24]	DGT[13]	Ours
FM	0.670/0.501	0.752/0.542	0.723/0.556	0.802/0.588	0.758/0.566	0.681/0.534	0.391/0.318	0.814/0.570	0.777/0.549	0.802/0.607
BC	0.716/0.524	0.746/0.519	0.775/0.570	0.841/0.612	0.769/0.542	0.734/0.550	0.515/0.372	0.842/0.585	0.867/0.614	0.833/0.623
MB	0.642/0.493	0.731/0.556	0.702/0.567	0.791/0.601	0.731/0.568	0.669/0.533	0.387/0.320	0.803/0.585	0.815/0.591	0.795/0.605
DEF	0.712/0.514	0.754/0.489	0.741/0.527	0.858/0.580	0.772/0.535	0.689/0.499	0.451/0.295	0.778/0.523	0.857/0.582	0.859/0.591
IV	0.772/0.551	0.728/0.515	0.766/0.554	0.837/ 0.593	0.777/0.554	0.732/0.557	0.515/0.401	0.792/0.532	0.838/ 0.573	0.828/ 0.599
IPR	0.756/0.520	0.794/0.529	0.828/0.567	0.848/ 0.579	0.785/0.546	0.782/0.557	0.471/0.348	0.853/0.559	0.856/ 0.573	0.825/ 0.573
LR	0.773/0.406	0.808/0.382	0.903/0.423	0.936/ 0.514	0.818/ 0.515	0.699/0.399	0.751/0.465	0.847/0.388	0.732/0.417	0.856/ 0.507
OCC	0.715/0.520	0.741/0.504	0.754/0.528	0.836/ 0.585	0.756/0.542	0.682/0.507	0.454/0.335	0.755/0.520	0.820/0.562	0.826/ 0.576
OPR	0.734/0.523	0.794/0.525	0.787/0.547	0.847/ 0.576	0.777/0.543	0.746/0.538	0.509/0.371	0.807/0.534	0.855/ 0.577	0.848/ 0.590
OV	0.594/0.446	0.685/0.488	0.633/0.497	0.794/ 0.565	0.692/0.508	0.592/0.452	0.558/ 0.384	0.676/0.474	0.753/0.533	0.749/ 0.553
SV	0.732/0.506	0.736/0.470	0.746/0.475	0.825/ 0.552	0.764/0.551	0.681/0.488	0.535/0.391	0.790/0.481	0.813/0.504	0.817/ 0.552
ALL	0.755/0.537	0.781/0.530	0.803/0.560	0.869/ 0.608	0.802/0.575	0.762/0.562	0.526/ 0.384	0.831/0.559	0.865/0.586	0.867/ 0.616

each frame so that the minimum side length of a bounding box is 32 pixels and fixed the side length of a searching window is $2\sqrt{wh}$, where w and h denote the width and height of bounding box [1]. In Eqs.(3,5), we set the restart term $\alpha = 0.8$. In Eq.(6), we set the scaling parameter $\sigma = 5$.

Table II: Tracking performance of our OWP (PR, SR, FPS) on different patch partitions on OTB100 dataset

Patches partition	PR/SR	FPS
6×6 (36 patches)	0.828/0.591	16.0
7×7 (49 patches)	0.858/0.604	14.4
8×8 (64 patches)	0.867/0.616	13.5
9×9 (81 patches)	0.859/0.609	12.1
10×10 (100 patches)	0.855/0.603	10.2

OTB100 benchmark dataset. This dataset contains 100 image sequences which are associated with different attributes, such as illumination variation, scale variation, occlusion and background clutters, etc [26]. Similar to many other works, we use both precision rate (PR) and success rate (SR) [27], [26] to measure the quantitative performances of different methods. The precision at the distance threshold of 20 pixels is used as the representative Precision Rare (PR), and the average success rate whose area under the success rate curve over all overlap thresholds is used as the representative Success Rate (SR).

Temple-Color benchmark dataset. This dataset contains 128 challenging image sequences of human, animals and rigid objects [25]. Each sequence in this dataset is annotated by its challenge factors and ground truth track, which are the same with work [26]. We use Precision Rate (PR) and Success Rate (SR) as the evaluations which are the same with work [26].

B. Comparison results

We compare our method with some state-of-the-art tracking methods including both deep and non-deep learning based tracking methods.

Comparisons with non-deep learning based trackers. We compare our tracker with some state-of-the-art non-deep learning based tracking methods including DGT [13], ReGLe [14], SOWP [1], MEEM [8], Staple [21] and LCT [20]. Note that, the tacker DGT [13], ReGLe [14] and SOWP [1] also employ weighted patch descriptor for object tracking and thus are most related with our tracking method. Figure 2 shows the comparison results on one-pass evaluation (OPE) using the distance precision rate (PR) and overlap success rate (SR) curves, respectively. Overall, we can note that our tracker generally obtains the best result in SR and the second best in PR. In particular, it achieves 0.2%/3.0%, 6.4%/5.6% performance gains in PR/SR over DGT [13] and SOWP [1], and 0.8% performance gain in SR over ReGLe [14] which are most related trackers with our work. Note that, our method performs obviously faster than related work DGT [13], ReGLe [1] and SOWP [1] because of closed-form solution computation of the proposed optimization model, as discussed before.

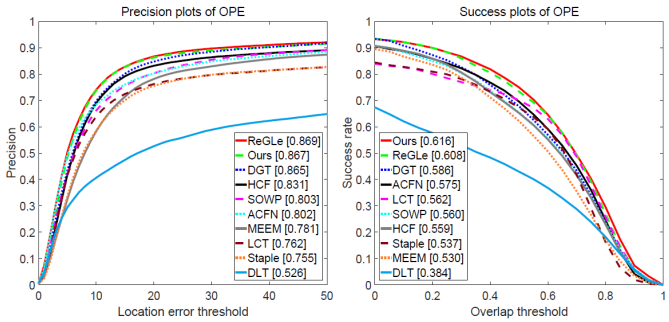


Figure 2: Evaluation results on OTB100 benchmark dataset. The representative score of PR/SR is presented in the legend.

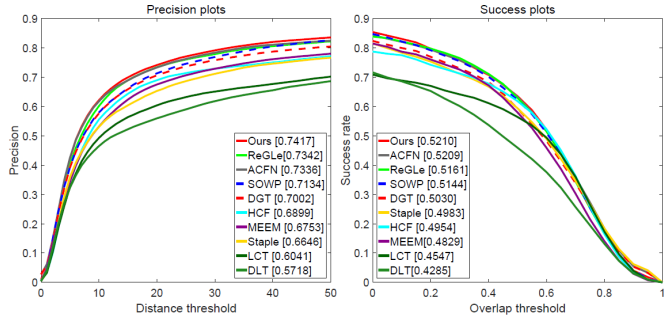


Figure 3: Evaluation results on TColor-128 benchmark dataset. The representative score of PR/SR is presented in the legend.

Comparisons with deep learning based trackers. In addi-

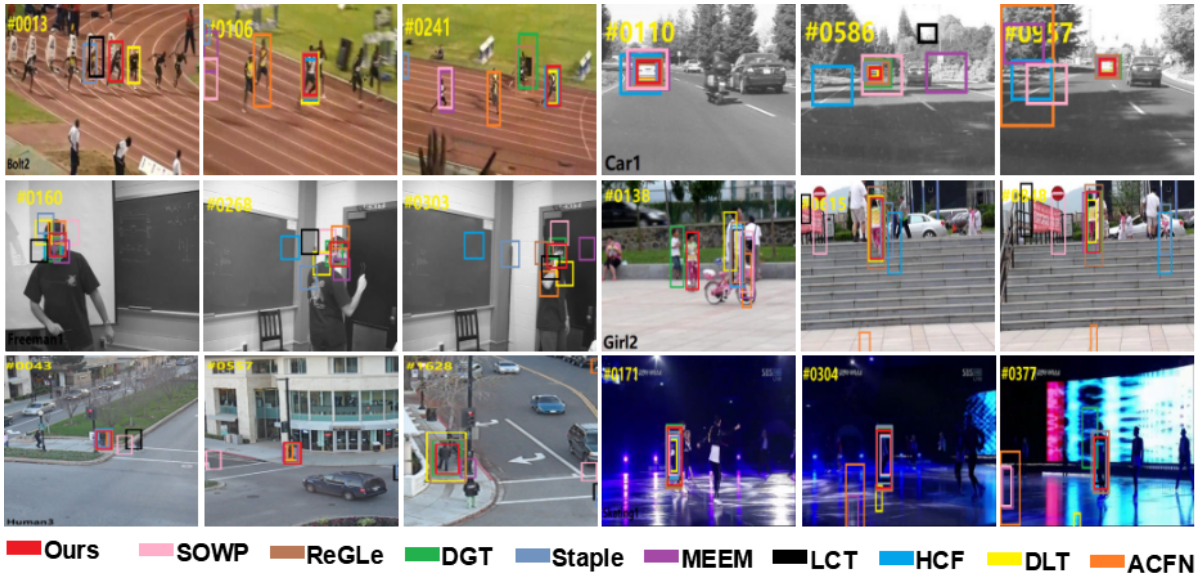


Figure 4: Tracking examples of different tracking methods on 6 challenging video sequences (from left to right and top to down are Bolt2, Car1, Freeman1, Girl2, Human3 and Skating1, respectively).

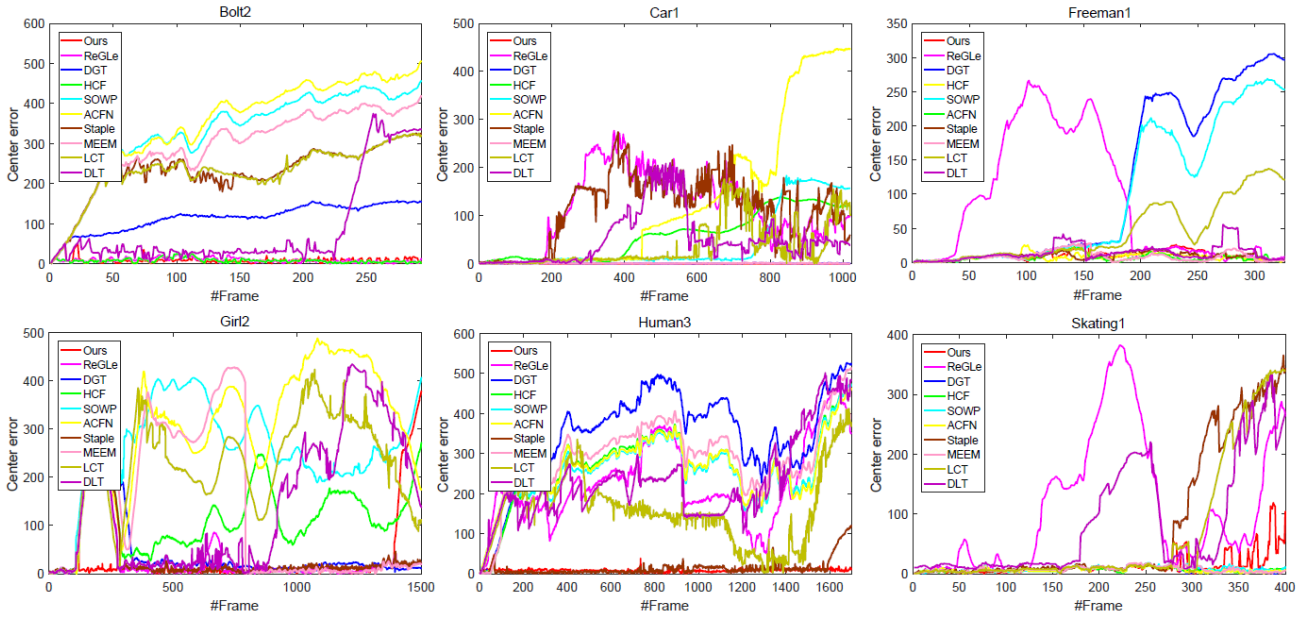


Figure 5: Comparison of center location errors (in pixels) on 6 challenging video sequences. Generally, our method obtains lower center location errors.

tional to non-deep learning tracking methods, we also compare our tracker with some other deep learning based tracking methods including ACFN [22], DLT [23] and HCF [24]. Figure 2 presents the comparison results which show that our tracker outperforms ACFN, DLT and HCF. Note that, comparing with deep learning based trackers, our tracker does not need large-scale annotated training samples and just uses the ground truth in the first frame to train our model and updates the model in subsequent frames.

Evaluation on different attributes. Table I shows the representative PR/SR results on 11 different attributes, respectively. Overall, one can note that the proposed tracking method get comparable results with other trackers on most challenging attributes in SR, which indicates the effectiveness of our method. Especially, our tracker obtains better performance on the videos of fast motion, motion blur, deformation, illumination variation, occlusion, out-of-plane rotation and scale variation, which further demonstrates the effectiveness of our OWP descriptor on suppressing the background effects and noises. Figure 4 shows some tracking results on some

challenging examples. Figure 5 shows the corresponding center location error [20]. The lower of center error value is, the more accurately the visual object is located. One can note that, our tracker locates the visual object more accurately on these challenging sequences.

Comparisons on Temple-Color dataset. In addition to OTB100 dataset, we further evaluate our method on Temple-Color benchmark dataset [25]. Figure 3 shows the results on this dataset. Overall, we can note that our tracker generally outperforms the other trackers and achieves the best performance on this benchmark dataset. Especially, our method achieves 2.83%/0.66%, 4.15%/1.80% and 0.75%/0.49% performance gains in PR/SR over SOWP, DGT and ReGLe while performs faster than these methods, which further demonstrates the effectiveness of our tracker.

VI. Conclusions

This paper proposes a novel Objectness Weighted Patch (OWP) descriptor for visual tracking problem. OWP aims to assign different objectness weights to the patches of bounding box to alleviate the influences of cluttered background and noises in visual tracking process. To compute the objectness of patches, we propose a general optimization model by integrating multiple cues including background, foreground and local consistency together. Moreover, the proposed model has a simple closed-form solution and thus can be computed efficiently. Extensive experiment on two standard benchmark OTB100 and TColor-128 have demonstrated the superior performance of our approach over other related tracking methods.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (61602001,61671018); Natural Science Foundation of Anhui Province (1708085QF139); Natural Science Foundation of Anhui Higher Education Institutions of China (KJ2016A020).

References

[1] H. U. Kim, D. Y. Lee, J. Y. Sim, and C. S. Kim, "Sowp: Spatially ordered and weighted patch descriptor for visual tracking," in *IEEE International Conference on Computer Vision ICCV*, 2015, pp. 3011–3019.

[2] Z. He, S. Yi, Y.-M. Cheung, X. You, and Y. Y. Tang, "Robust object tracking via key patch sparse representation," *IEEE transactions on cybernetics*, vol. 47, no. 2, pp. 354–364, 2017.

[3] B. Zhang, Z. Li, A. Perina, A. Del Bue, V. Murino, and J. Liu, "Adaptive local movement modeling for robust object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 7, pp. 1515–1526, 2017.

[4] C. Dorin, R. Visvanathan, and M. Peter, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 564–575, 2003.

[5] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M. M. Cheng, S. L. Hicks, and P. H. S. Torr, "Struck: Structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 2096–2109, 2016.

[6] S. He, Q. Yang, R. W. H. Lau, J. Wang, and M. H. Yang, "Visual tracking via locality sensitive histograms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2427–2434.

[7] Y. Yuan, H. Yang, Y. Fang, and W. Lin, "Visual object tracking by structure complexity coefficients," *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1125–1136, 2015.

[8] J. Zhang, S. Ma, and S. Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," in *Proc. of the European Conference on Computer Vision (ECCV)*, vol. 8694, 2014, pp. 188–203.

[9] F. Liu, C. Gong, T. Zhou, K. Fu, X. He, and J. Yang, "Visual tracking via nonnegative multiple coding," *IEEE Transactions on Multimedia*, vol. 19, no. 12, pp. 2680–2691, 2017.

[10] S. Duffner and C. Garcia, "Pixeltrack: A fast adaptive algorithm for tracking non-rigid objects," in *IEEE International Conference on Computer Vision*, 2013, pp. 2480–2487.

[11] F. Yang, H. Lu, and M. H. Yang, "Robust superpixel tracking," *IEEE Transactions on Image Processing*, vol. 23, pp. 1639–1651, 2014.

[12] R. Yao, S. Xia, Z. Zhang, and Y. Zhang, "Real-time correlation filter tracking by efficient dense belief propagation with structure preserving," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 772–784, 2017.

[13] C. Li, L. Lin, W. Zuo, and J. Tang, "Learning patch-based dynamic graph for visual tracking," in *AAAI*, 2017.

[14] C. Li, X. Wu, Z. Bao, and J. Tang, "Regle: Spatially regularized graph learning for visual tracking," in *ACM MM*, 2017, pp. 252–260.

[15] M. Yang, X. Ruan, H. Lu, L. Zhang, and C. Yang, "Saliency detection via graph-based manifold ranking," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2013, pp. 3166–3173.

[16] S. Lu, V. Mahadevan, and N. Vasconcelos, "Learning optimal seeds for diffusion-based salient object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2790–2797.

[17] Z. He, B. Jiang, Y. Xiao, C. Ding, and B. Luo, "Saliency detection via a graph based diffusion model," in *International Workshop on Graph-Based Representations in Pattern Recognition*, 2017, pp. 3–12.

[18] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu, "Automatic multimedia cross-modal correlation discovery," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 653–658.

[19] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2814–2821.

[20] C. Ma, X. Yang, C. Zhang, and M. H. Yang, "Long-term correlation tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5388–5396.

[21] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1401–1409.

[22] J. Choi, H. J. Chang, S. Yun, T. Fischer, Y. Demiris, and Y. C. Jin, "Attentional correlation filter network for adaptive visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[23] W. Naiyan and Y. Dit-Yan, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems 26*, 2013, pp. 809–817.

[24] C. Ma, J. B. Huang, X. Yang, and M. H. Yang, "Hierarchical convolutional features for visual tracking," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3074–3082.

[25] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5630–5644, 2015.

[26] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 37, pp. 1834–1848, 2015.

[27] 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, pp. 583–596, 2015.