Visual Object Tracking Using PCA Correlation Filters

Yinxia Lu, Zhenghua Zhou and Jianwei Zhao*
College of Sciences, China Jiliang University, Hangzhou 310018, China
* Corresponding author

***Abstract*—Accurate translation and robust scale estimation are two challenging problems for visual object tracking. Many existing trackers use some feature extraction methods and the exhaustive scale methods to solve above two problems, respectively. This paper continues to discuss above problems in the tracking-by-detection framework. It proposes an efficient tracker that applies Principal-Component-Analysis (PCA) features to learn the PCA correlation filters, which predicts the location of the target more accurately. Furthermore, our proposed tracker keeps the good performance for the scale variation by using an accurate scale estimation method. Experimental results show that our proposed tracker has a better accuracy for predicting the location of the target and a higher percent in the average overlap precision than some other methods on the 30 benchmark sequences with scale variation.**

Keywords—visual tracking; correlation filter; principal component analysis filter

I. INTRODUCTION

Visual tracking has wide applications in real life, such as motion analysis, activity recognition, visual surveillance, traffic monitoring, and so on. Although many excellent tracking methods have been proposed in recent years, there still exist a lot of problems for overcoming, such as deformation, partial occlusion, motion blur, and so on.

In general, visual tracking mainly consists of two main issues. One is to estimate the location of the target in each frame of an image sequence. The other is to select an optimal scale of the tracking box for the object. Therefore, it is urgent and challenge to solve these two problems well, especially confronting the large scale variation in some complex video sequences.

Recently, some detection-based tracking methods [1–3] have shown their good tracking effect. These tracking methods regard the task of target localization as a problem of classification. The decision boundary is obtained by means of learning a discriminative classifier online with some image patches from the target and the background. For example, Bao et al. [4] proposed a tracker that found an adaptive correlation filter by minimizing the output sum of squared error (MOSSE). Henriques et al. [2] proposed a tracker that worked by learning a kernelized least-squares classifier of the target appearance. These methods improve the development of tracking methods based on discriminative correlation filters. The rule of discriminative-correlation-filter-based trackers is to train a correlation filter online with a set of training image patches from the object and background. Then the target location can be determined by the optimal value of the learned correlation filter acting on the image patch sequence. Subsequently, Danelljan et al. [5] proposed a tracker, named DSST tracker, by means of learning the separate filters for translation and scale estimation based on MOSSE. DSST tracker has shown the commendable performance and a high tracking speed.

Generally, it is an important step to design a good feature extraction method for training the correlation filters in those trackers. DSST tracker [5] uses the HOG features for training the translation filter and concatenates it with the usual image intensity features. In this paper, we propose an efficient tracking method based on the DSST tracker with a simple and efficient feature extraction, Principal Component Analysis (PCA), for training the filters. The proposed tracker has an accurate tracking result in detecting the location of target and still keeps the adaption for the scale variation of the target in the process of tracking.

II. OUR PROPOSED TRACKER

In this section, we describe our proposed tracking method based on the DSST tracker and PCA correlation filters. Subsection 2.1 proposes a simple and efficient feature extraction, Principal Component Analysis (PCA), to extract the feature of the patch for training the PCA correlation filters in our tracker. Subsection 2.2 gives the process of training PCA correlation filters for the multidimensional features. Subsection 2.3 describes the scale estimation method.

A. Feature Extraction

In our proposed tracker, we will use some features of the target appearance as samples for training the correlation filters. In this subsection, we use PCA to extract the multi-dimension features of the target patch for training the PCA correlation filters in our tracker.

Given an image patch $z \in \mathbb{R}^{m\times n}$ of the target object, we can get $q := (m-k_1+1) \times (n-k_2+1)$ patches $X_1, X_2, \ldots, X_q \in \mathbb{R}^{k_1\times k_2}$ by a sliding window of size $k_1 \times k_2$ on the object patch. Let x_i be the corresponding vectorization of the patch X_i, $i = 1, 2, \ldots, q$, and put them together, we can get a matrix $X = [x_1, x_2, \ldots, x_q]^{\mathrm{T}} \in R^{q\times k_1 k_2}$.

Assuming that the number of correlation filters in our tracker is d , solve the eigenvalues of the covariance matrix XX^{T} and rearrange them from large to small. Select d largest eigenvalues and get the corresponding eigenvectors. Let $q_l(XX^{\mathrm{T}})$ means the l -th principal eigenvector, $l = 1, 2, \ldots, d$. Put

$$W^l = mat_{k_1 \times k_2}(q_l(XX^{\mathrm{T}})) \in R^{k_1 \times k_2}, \quad l = 1, 2, \ldots, d \quad (1)$$

where $mat_{k_1 \times k_2}(v)$ is an operation that maps $v \in R^{k_1 \times k_2}$ to a matrix of size $k_1 \times k_2$. We call the matrix W^l the PCA filter.

With these PCA filters, we can get the features for each patch z by the following formula

$$z^l = z * W^l, \quad l = 1, 2, \ldots, d \quad (2)$$

where $*$ denotes the circular correlation.

B. Mutil-channel PCA Correlation Filters

In this subsection, we give the process of training PCA correlation filters with the multi-dimensional features. For the rectangular patch of the target f and a desired correlation output g , extract the features $\{f^l\}_{l=1}^d$ of f with the feature extraction method in subsection 2.1. The goal of our tracker is to find the optimal PCA correlation filters $\{h^l\}_{l=1}^d$. They can be achieved by minimizing the following l^2 -error of the correlation response to the desired correlation output g :

$$\underset{\{h^l\}_{l=1}^d}{\arg\min} \left\| \sum_{l=1}^d h^l * f^l - g \right\|^2 + \lambda \sum_{l=1}^d \left\| h^l \right\|^2, \quad (3)$$

where g is the desired correlation output constructed as a two-dimensional Gaussian function with its peak located at the target centre, $*$ denotes the circular correlation, and λ is a positive regularization parameter.

We solve the optimal problem (3) on the frequent area with discrete Fourier transform (DFT). Then the DFT of h^l in (3) can be obtained by

$$H^l = \frac{\overline{G}F^l}{\sum_{k=1}^d \overline{F^k}F^k + \lambda}, \quad l = 1, 2, \ldots, d, \quad (4)$$

where H^l, G, F^l are the corresponding DFT of h^l, g, f^l , respectively, the bar $\overline{G}$ means the complex conjugation of G , and the multiplications and divisions are performed in point-wise.

As shown in [9], the optimal correlation PCA filters can be obtained by minimizing the output error over all training patches. However, it requires to solve a $d \times d$ linear system of equations per pixel. Therefore, we compute a robust approximation by updating the numerator A_t^l and the denominator B_t of the correlation filter H_t^l as

$$A_t^l = (1-\eta)A_{t-1}^l + \eta \overline{G_t} F_t^l \quad (5)$$

and

$$B_t = (1-\eta)B_{t-1} + \eta \sum_{k=1}^d \overline{F_t^k} F_t^k \quad (6)$$

where the scalar η is a learning rate, A_t^l, B_t , and H_t^l are the numerator, the denominator, and the correlation filter at frame t .

For a test rectangular region z centered around the predicted target location, extract the features $\{z^l\}_{l=1}^d$ with the feature extraction method in subsection 2.1. Then its correlation score y can be computed by

$$y = F^{-1}\left\{ \frac{\sum_{l=1}^d \overline{A_t^l} z^l}{B_t + \lambda} \right\}. \quad (7)$$

The estimate location of the target in the new frame is obtained by finding the maximum correlation score y .

C. Extracting Appropriate Scale

With Subsection 2.1 and Subsection 2.2, we can get a precise location of the target in a new frame by learning a multi-channel correlation PCA filters from a single sample f . In this step, we get a two-dimensional translation PCA correlation filters. However, the size of the filter is not fixed. We may suppose that the scale filter is one dimension. Therefore, the same method can be used to learn a one-dimensional scale estimation filter. Although the scale estimation of the tracker increases a little computational cost, we can get more accurate target location and size.

We confirm the size of the target by learning an extra scale filter. To get the training sample, we extract features using variable patch sizes centered around the target. Assume $P \times Q$ denotes the target size in the current frame and s be the size of the scale filter. In the light of each $n \in \left\{ \left\lfloor -\frac{s-1}{2} \right\rfloor, \ldots, \left\lfloor \frac{s-1}{2} \right\rfloor \right\}$, we extract an image patch J_n of size $a^n P \times a^n Q$ centred around the target. Here, a denotes the scale factor between feature layers. At scale level n, the value $f(n)$ about the training sample is set to the d -dimensional feature descriptor of J_n .

Finally, the scale filter with the new sample is updated by formulas (5) and (6). In this case, we use a one-dimensional Gaussian as the desired correlation output g. In general, the size change of the target between two adjacent frames is smaller than the position change. Therefore, we first compute the position. Afterwards, the scale filter is applied at the new target location.

Algorithm 1 provides a brief outline of our proposed tracker.

```
**Algorithm 1** Proposed tracking method: iteration at frame $t$.
**Require:** Image $I_t$, previous target position $P_{t-1}$ and scale $s_{t-1}$, translation model $A_{t-1}^{trans}$, $B_{t-1}^{trans}$ and the scale model $A_{t-1}^{scale}$, $B_{t-1}^{scale}$.
**Ensure:** Estimated the target position $P_t$ and scale $s_t$, updated the translation model $A_t^{trans}$, $B_t^{trans}$ and the scale model $A_t^{scale}$, $B_t^{scale}$.
    **Feature Extraction**
1: Divide the target image into patches, and use them to construct a matrix;
2: Apply PCA method to the matrix to construct PCA filters $W^l$;
3: With these PCA filters, get the features of the target patch;
    **Position Detection**
4: For a translation sample $z_{trans}$ from $I_t$ at the target position $P_{t-1}$ and scale $s_{t-1}$, extract its PCA features.
5: Compute the translation correlation $y_{trans}$ using PCA features, $A_{t-1}^{trans}$, and $B_{t-1}^{trans}$ in (7).
6: Let $P_t$ be the target location that minimizes $y_{tran}$.
    **Scale Detection**
7: For a sacle sample $z_{trans}$ from $I_t$ at the target position $P_t$ and scale $s_{t-1}$, extract its PCA features.
8: Compute the scale correlation $y_{scale}$ using PCA features, $A_{t-1}^{scale}$, and $B_{t-1}^{scale}$ in (7).
9: Let $s_t$ be the target scale that minimizes $y_{scale}$.
    **Updating**
10: Update the translation model $A_t^{trans}$, and $B_t^{trans}$ in (7).
11: Update the scale model $A_t^{scale}$, and $B_t^{scale}$ in (7).
```

III. Experimental Results

In this section, we compare our tracker with some other state-of-the-art trackers, such as DSST [5], IVT [6], MTT [7], LOT [8], OAB [10], LSK [11], MIL [12], SBT [13], CT [3], KMS [14], SMS [15] and DFT [16]. All the experiments are performed in Matlab2015b on a computer with Intel Xeon CPU E5-1620 v3 @ 3.50GHz.

To evaluate the performance of our proposed tracker, we carried out some experiments on 30 available challenging image sequences with 11 attributes: low resolution (LR), in-plane rotation (IPR), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), background clutter (BC), illumination variation (IV), motion blur (MB), fast motion (FM), and out of view (OV).

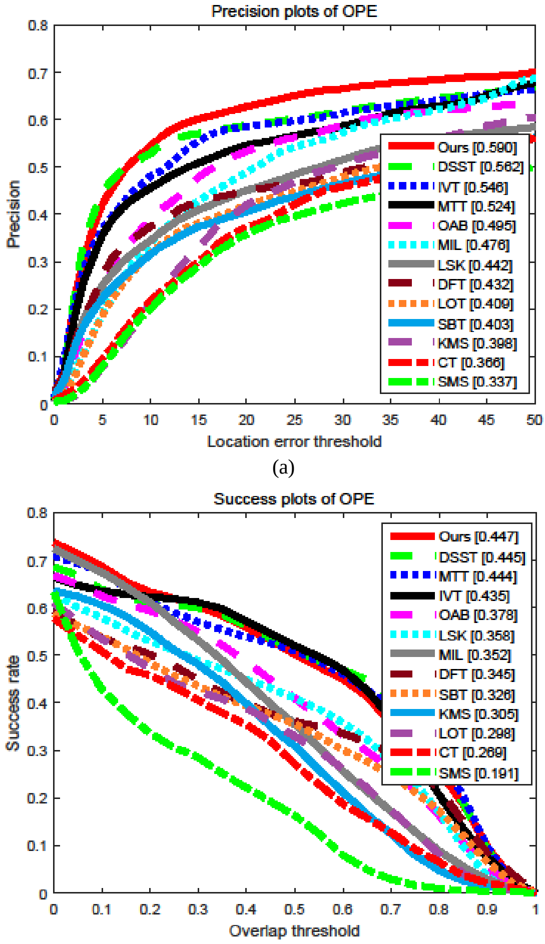
We set the regularization parameter $\lambda = 0.01$ and the learning rate $\eta = 0.025$. The filter size $P \times Q$ is set to twice the initial target size. The standard deviations for desired correlation output is set to 1/16 of the target size for translation filter and 1.5 in the scale filter. The scale number is 33 with a scale factor of $a = 1.02$. The number of convolution filter is set to 50 and the patch size is 9 × 9. We keep the same parameters values for all the sequences. The 51-dimensional feature is grayscale map of sample. 33 different sizes of scale training samples extracted from each frame are resized to the same size, which ensures a maximum feature descriptor length of 992. Finally, the extracted features are always multiplied by a Hann window, as described in [4].

We employ the success plot, the precision plot, and the centre location error (CLE) as the quantitative evaluations. The precision plot shows the percentage of frames whose center locations are in the given threshold to the ground truth. The success plot is based on an overlap ratio defined as

$$S = Area(r_t \cap r_g) / Area(r_t \cup r_g), \qquad (8)$$

where r_t is the bounding box of the tracking result and r_g denotes that of the groundtruth, $\cap$ and $\cup$ represent the intersection and union of two regions, respectively. We count the number of successful frames with overlap ratio S greater than the predefined t_0 and calculate the ratio of successful frames throughout all thresholds $t_0 \in [0,1]$ to draw the success plot. We employ the area under the curve (AUC) of each success plot and precision plot to rank the trackers. CLE is the average value of the Euclidean distance between the groundtruth and the estimated centre location.

Figure 1 shows the success plot and precision plot of 13 trackers in a benchmark database. We rank the trackers using their corresponding representative scores with a threshold of 20 pixels. As observed from Figure 1, our track ranks the first in both precision plot and success plot. In the precision plot of OPE, our tracker gets a score 0.590 that outperforms the DSST tracker with 0.028. In the success plot of OPE, our tracker gets a score 0.447 that outperforms the DSST tracker with 0.002. The experimental results show that our tracker has a good performance in finding the location of the target center, and a little improvement in detecting the scale of target.

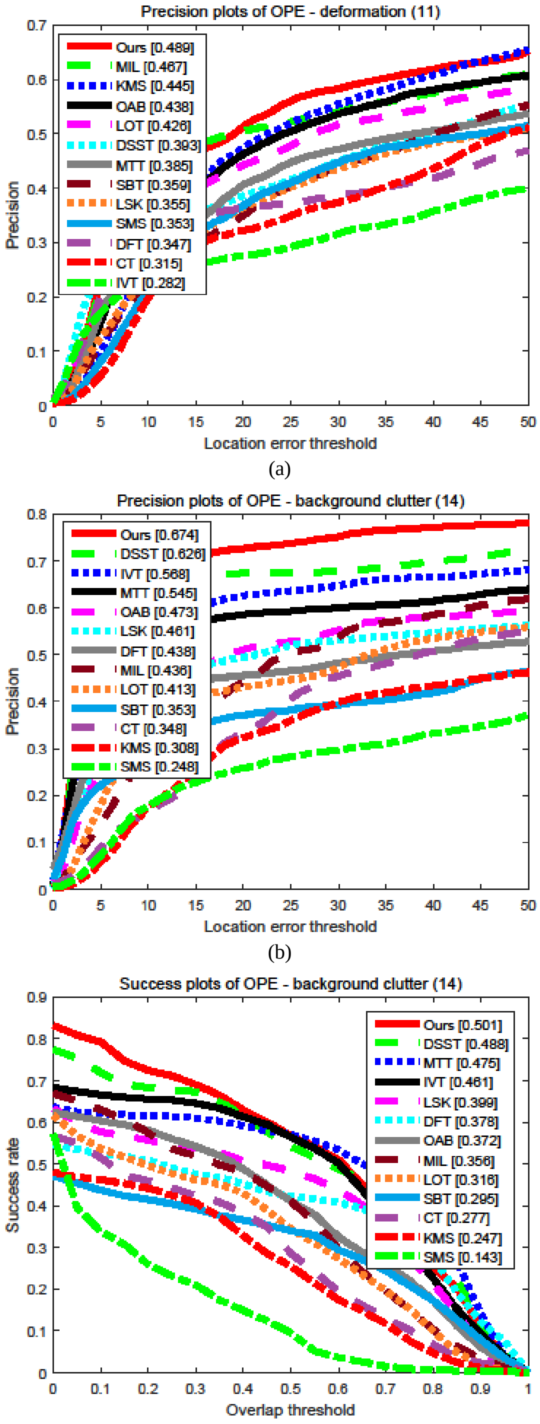


(a)

(b)

FIGURE I. THE PRECISION PLOT AND SUCCESS PLOT OF OUR TRACKER WITH OTHER 12 TRACKERS.

Figure 2 shows the precision plot of OPE on the attribution deformation and background clutter, and the success plot of OPE on the attribution background clutter for 13 trackers in a benchmark database. As observed from Figure 2, our tracker ranks the first in both precision plot and success plot on some attributions. Especially on the attribution background clutter, our tracker gets a precision plot of OPE 0.674 that outperforms the DSST tracker with 0.048. And our tracker gets a success plot 0.501 that outperforms the DSST tracker with 0.013.



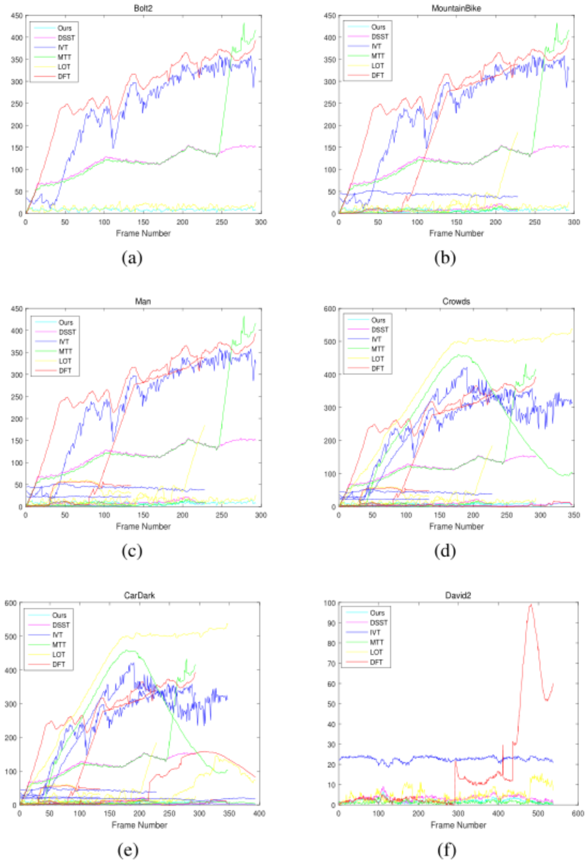
(a)

(b)

(c)

FIGURE II. OUR ADVANTAGES IN SOME ATTRIBUTES.

Figure 3 shows the Euclidean distance of the detection center position and the standard position of 6 trackers over 6 sequences: Bolt2, MountainBike, Man, Crowds, CarDark, and David2. As observed from Figure 3, our tracker always has small Euclidean distances among 6 trackers, which implies that our tracker can tracking the target very well.



(a) (b) (c) (d) (e) (f)

FIGURE III. THE EUCLIDEAN DISTANCE OF THE DETECTION CENTER POSITION AND THE STANDARD POSITION OF 6 TRACKERS OVER 6 SEQUENCES: BOLT2, MOUNTAINBIKE, MAN, CROWDS, CAR-DARK, AND DAVID2.

Figure 4 shows the sampled tracking results of 6 trackers over 6 sequences.

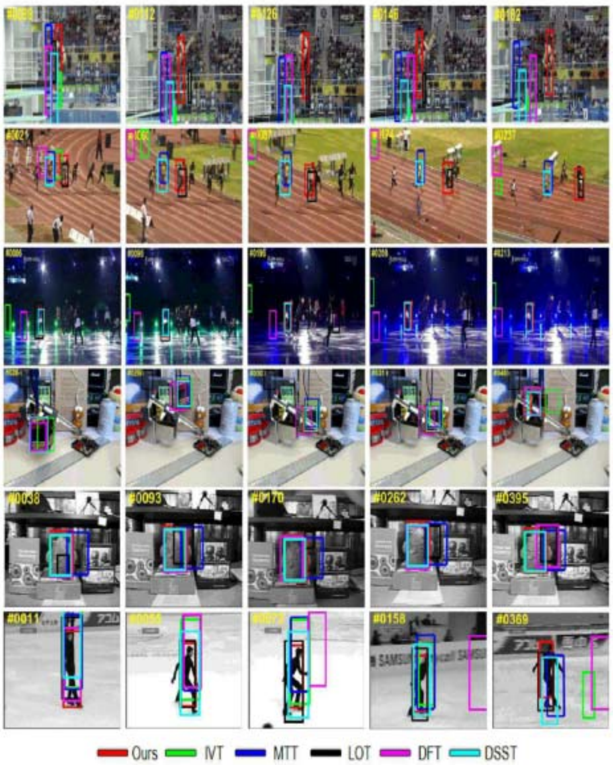


FIGURE IV. A SIMPLE COMPARISON OF OUR TRACKER (IN RED) WITH OTHER FIVE TRACKERS, IVT (IN GREEN), MTT (IN BLUE), LOT (IN BLACK), DFT (IN ROSE RED), AND DSST (IN SKY BLUE).

Table 1 shows the precision of 6 trackers on 30 sequences. As observed from Table 1, our tracker has the best precision on 24 sequences among 5 trackers. On the sequences: Car2, CarDark, Fish, Coupon, and Car1, all 5 trakers have unsatisfied tracking results. The reason is that the objects in these sequences are affected by appearance variation and illumination variation.

TABLE I. THE PRECISION OF 6 TRACKERS ON 30 VIDEO SEQUENCES (%).

Precision	**Diving**	**Dudek**	**David2**	**BlurFace**	**BlurBody**	**Bolt2**	**Mountain Bike**	**Football1**	**Panda**	**Man**
Ours	100	100	100	100	100	100	100	100	100	100
DSST	100	100	100	100	100	100	100	100	100	100
CT	98.6	84.7	83.3	79.6	63.8	54.9	52.2	48.8	45.9	45.2
LOT	100	100	100	98.9	87.7	74.8	69.3	66.1	61.3	61.1
DFT	100	100	100	100	93.3	86.1	84.3	80.4	60.7	56
Precision	**Crowds**	**Box**	**Kitesurf**	**MotorRolling**	**Singer1**	**Human7**	**Football**	**Jump**	**Freeman1**	**Car4**
Ours	55.3	51.4	45.2	44.7	44.4	38.7	37.9	34.4	33.8	10.7
DSST	51.4	45.2	40.9	39	38.7	38	29.2	23.7	9.02	3.84
CT	24.6	21.3	21.1	17.1	15.4	11.2	5	4.19	2.3	1.86
LOT	23.1	22.5	18.5	16.4	12.3	11.3	10.7	10.5	7.14	5.4
DFT	28.2	26.4	23.2	23.1	18.8	18.5	13.4	12.3	10.8	10.7
Precision	**Sylvester**	**Skating1**	**Skater2**	**Car24**	**Walking**	**Car2**	**CarDark**	**Fish**	**Coupon**	**Car1**
Ours	100	99.6	83.8	79.8	65.7	4.37	0	0	0	0
DSST	100	96.9	84.2	79.8	57.8	1.71	0	0	0	0
CT	38.3	37.2	36.8	32.5	31.1	1.83	0.865	0.599	0.509	0.372
LOT	53.7	51.2	44	39.4	27	5.39	5.01	3.7	3.66	2.02
DFT	54.5	47.3	41.7	35.1	31.2	8.55	7.19	4.27	3.45	1.71

Table 2 describes the median of success, precision, CLE, and frame per second (FPS) of our proposed tracker with DSST, CT, LOT, and DFT trackers. As observed from Table 2, our tracker has the best median precision and median CLE, a second median success, and the third median FPS among 5 trackers. The reason for the slower tracking speed than DSST is that our tracker needs to extract the PCA features.

TABLE II. THE MEDIAN OF SUCCESS, PRECISION, CLE, AND FRAME PER SECOND (FPS) OF OUR PROPOSED TRACKER WITH DSST, CT, LOT, AND DFT TRACKERS.

Method	Median success	Median precision	Median CLE	Median FPS
Ours	36.5	**65.7**	**27.1**	19.27
DSST	**39.2**	57.8	34.7	38.15
CT	20.8	31.1	81.7	**49.87**
LOT	21	27	78.3	0.49
DFT	25.6	31.2	58.7	10.73

IV. CONCLUSIONS

In this paper, we have proposed an efficient tracker that applies Principal-Component-Analysis (PCA) features to learn the PCA correlation filters, which predicts the location of the target more accurately. Furthermore, our proposed tracker keeps the good performance for the scale variation by using an accurate scale estimation method. Experimental results showthat our proposed tracker has a better accuracy for predicting the location of the target and a higher percent in the average overlap precision than some other methods on the 30 benchmark sequences with scale variation.

ACKNOWLEDGMENT

This work was funded by the National Natural Science Foundation of China (61571410) and the Zhejiang Provincial Nature Science Foundation of China (LY18F020018).

REFERENCES

[1] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: structured output tracking with kernels," International Conference on Computer Vision, vol. 23, no. 5, pp. 263-270, 2011.

[2] C. Rui, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," European Conference on Computer Vision, vol. 7575, no.1, pp. 702-715, 2012.

[3] K. Zhang, L. Zhang, and M. H. Yang, "Real-time compressive tracking," European Conference on Computer Vision, vol. 7574, no.1, pp. 864-877, 2012.

[4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters. Computer Vision & Pattern Recognition, vol. 119, no. 5, pp. 2544-2550, 2010.

[5] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," British Machine Vision Conference, vol. 65, pp. 1-65, 2014

[6] 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.

[7] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," Computer Vision & Pattern Recognition, vol. 157, no. 10, pp. 2042-2049, 2012.

[8] S. Avidan, D. Levi, A. Barhillel, and S. Oron, "Locally orderless tracking," International Journal of Computer Vision, vol. 111, no. 2, pp. 213-228, 2012.

[9] H. Galoogahi, T. Sim, and S. Lucey, "Multi-channel correlation filters," IEEE International Conference on Computer Vision, vol. 2014, pp. 3072-3079, 2014

[10] G. Helmut, "Real-time tracking via on-line boosting," Proceeding of British Machine Vision Conference, vol. 1, pp. 47-56, 2006.

[11] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and K-selection," IEEE Conference on Computer Vision & Pattern Recognition, vol. 3619, no. 7, pp. 1313-1320, 2011

[12] B. Babenko, M. H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," IEEE Conference on Computer Vision & Pattern Recognition, vol. 33, no. 8, pp. 983-990, 2009.

[13] J. Sobreques, J. Espinasa, and J. Cebria, "Beyond semi-supervised tracking: tracking should be as simple as detection, but not simpler than recognition," IEEE International Conference on Computer Vision, vol. 53, no. 4, pp. 1409-1416, 2009.

[14] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564-575, 2003.

[15] R. Collins, “Mean-shift blob tracking through scale space,” IEEE Computer Society Conference on Computer Vision, vol. 2, pp. 234-240, 2003.

[16] L. Sevilla-Lara, and E. Learned-Miller, “Distribution fields for tracking,” IEEE Conference on Computer Vision & Pattern Recognition, vol. 157, no. 10, pp. 1910-1917, 2012.