COMP-LOP: COMPLEX FORM OF LOCAL ORIENTATION PLANE FOR OBJECT TRACKING

Miaobin Cen and Cheolkon Jung

School of Electronic Engineering, Xidian University, Xian 710071, China
zhengzk@xidian.edu.cn

ABSTRACT

In this paper, we propose complex form of local orientation plane (Comp-LOP) for object tracking. Comp-LOP is a simple but effective descriptor, which is robust to occlusion for object tracking. It effectively considers spatiotemporal relationship between the target and its surrounding regions in a correlation filter framework by the complex form, which successfully deals with the heavy occlusion problem. Moreover, scale estimation is performed to treat target scale variations for improving tracking accuracy. Besides, an appropriate size of the search region is determined without any manual adjustment by computing the entropy ratio of the target to background. Experimental results show that the proposed method achieves good performance in object tracking in terms of efficiency, accuracy, and robustness.

***Index Terms*—** Object tracking, correlation filter, complex form, entropy, occlusion, scale estimation.

1. INTRODUCTION

Object tracking aims at estimating the location of target in a video sequence. It is one of the most fundamental problems in computer vision with numerous applications such as motion-based recognition, automated surveillance, and human-computer interaction [1]. Up to now, a lot of tracking methods have been achieved by researchers [2–6]. Recently, correlation filters [7–11] have good performance in object tracking. In [7], Circulant Structure with Kernels (CSK) investigated a dense sampling strategy and the circulant structure for visual tracking. The CSK tracker [7] built only on illumination intensity features, and was further improved by using HOG in the Kernelized Correlation Filter (KCF) tracker [8]. In [9], Danelljan et al. extended the CSK tracker using color attributes or color names (CN), and proposed an adaptive low-dimensional variant of color attributes. Because this method introduces Fast Fourier Transformation (FFT) and the circulant structure like dense sampling, it has a high speed and an impressive accuracy. However, if the search region is large, this method needs much computation, which is not effective in dealing with the scale problem. Thus, we adopt correlation filters in this work due to their competitive performance.

In this paper, we propose complex form of local orientation plane (Comp-LOP) for object tracking. We utilize the entropy to find an appropriate search region as well as provide a novel scale update scheme for the scale problem. First, we select an appropriate search region and extract Comp-LOP in the region. Comp-LOP effectively considers spatiotemporal relationship between the target and its surrounding regions in a correlation filter framework by the complex form, and thus successfully deals with the occlusion problem. Then, we use FFT for response map (gaussian distribution) and features respectively and get their quotient A. Next, we extract Comp-LOP in the next frame and compute the target position with A by finding a position with the maximum response. Finally, we update scale and model of the target object. Fig. 1 illustrates the basic flow of the proposed object tracking. Compared with existing methods, main contributions of this work are as follows: 1) We propose a simple but effective descriptor for object tracking, i.e. Comp-LOP. Comp-LOP effectively considers the spatiotemporal relationship between the target and background, and thus successfully deals with the occlusion problem; 2) Given a query, we find an appropriate search region for object tracking based on the entropy to effectively utilize the target and background information; 3) We provide a scale update scheme to make the tracker more robust by comparing the responses of the consecutive frames.

2. CORRELATION TRACKING

Typical correlation trackers [7–11] learn a discriminative classifier to estimate the location of target by searching for the maximum value of the correlation response map. The classifier is trained from a single gray scale patch x of size $M \times N$ in an image centred around the target. Each cyclic shift $x_{m,n}$, $(m,n) \in \{0,1,...,M-1\} \times \{0,1,...,N-1\}$ as the training examples for the classifier. They are labelled using a gaussian function $y(m,n)$, $(m,n) \in \{0,1,...,M-1\} \times \{0,1,...,N-1\}$ and thus the classifier is generated by

This work was supported by the National Natural Science Foundation of China (No. 61271298) and the International S&T Cooperation Program of China (No. 2014DFG12780).

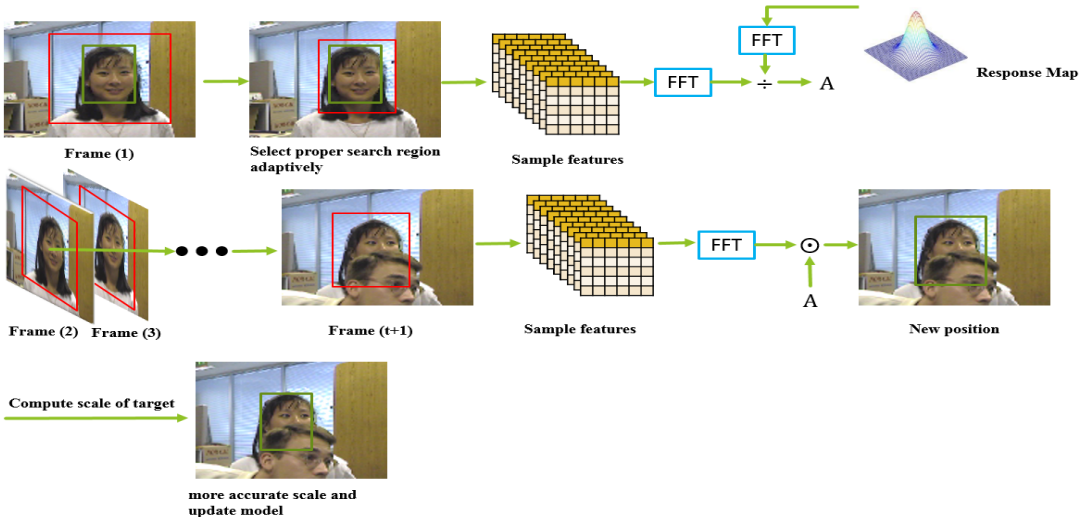

Fig. 1. Basic flow of the proposed method. Red and green rectangles represent the search region and target, respectively. Response map follows Gaussian distribution in (3). FFT is the Fast Fourier Transform.

minimizing the cost function (1) over w as follows:

$$\mathbf{w} = \arg\min_{\mathbf{w}} \sum_{m,n} |\phi(x_{m,n}) \cdot \mathbf{w} - y(m,n)|^2 + \lambda |\mathbf{w}|^2 \quad (1)$$

where ϕ denotes the mapping function to a kernel space; and $\lambda \geq 0$ is the regularization parameter. (1) is minimized by FFT to $w = \sum_{m,n} a(m,n)\phi(m,n)$, where the coefficient a is defined as follows:

$$A = \mathcal{F}(a) = \frac{\mathcal{F}(y)}{\mathcal{F}(\phi(x) \cdot \phi(x)) + \lambda} \quad (2)$$

where $\mathcal{F}$ is FFT; $y(m,n)$ is 1 near the target center position ($M/2$,$N/2$), and decreases to 0 as the distance increases:

$$y(m,n) = exp(-((m - M/2)^2 + (n - N/2)^2)/\sigma^2) \quad (3)$$

where σ is a scale parameter for the target size. Thus, the tracking task is to compute the response map on an image patch z in the next frame in the search region $M \times N$ as follows:

$$\hat{y} = \mathcal{F}^{-1}(A \odot \mathcal{F}(\phi(z) \cdot \phi(\hat{x}))) \quad (4)$$

where $\hat{x}$ is the learned target model and $\odot$ denotes the element-wise product. Thus, the new center position of the target is to find the position with the maximum value of $\hat{y}$.

3. PROPOSED METHOD

3.1. Complex Form-Local Orientation Plane

Since the complex form describes the orientation, e.g. $e^{\frac{\pi}{4}i}$ indicates that the orientation is $45°$, we adopt it to extract features and get a precise direction relationship between pixels.

Fig. 2 shows the whole process of the proposed feature extraction. As shown in the figure, we scan the whole search region without overlapping with a size of 3x3 block in the current frame t, and normalize each block. Then, we subtract the other 8 pixels from the center pixel of each block. Next, we add orientations into pixels except the center pixel to make them be complex form. Finally, we get the feature which consists of pixels at the same position from each block (total 9 channels). These 9 channels have local information because each channel represents one direction feature based on all small blocks. At the same time, to obtain the holistic relationship, we move the channel which is composed of pixels at the center position in each direction, i.e. 8 directions of $0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°$, and obtain other 8 channels. Then, we get the difference between these 8 channels and the center channel. Thus, we get the other 8 difference channels. These 8 channels have holistic information because each channel has a strong relationship with the center channel based on direction and difference. These 17 channels (the first 9 channels and the last 8 channels) have the local and holistic spatial relationship, and are connected by the center channel. Finally, we get the difference between the channels from the current frame and the previous frames to estimate the temporal relationship between consecutive frames. That is, each frame has total 34 channels. Due to the spatiotemporal information, the 34 channels are very effective in dealing with the occlusion problem in object tracking.

3.2. Appropriate Search Region

In conventional methods, the search region is manually adjusted, which is hard to find a suitable size in different sequences [6, 8, 12]. In this work, we utilize the entropy to automatically select its proper size. First, we initialize the large

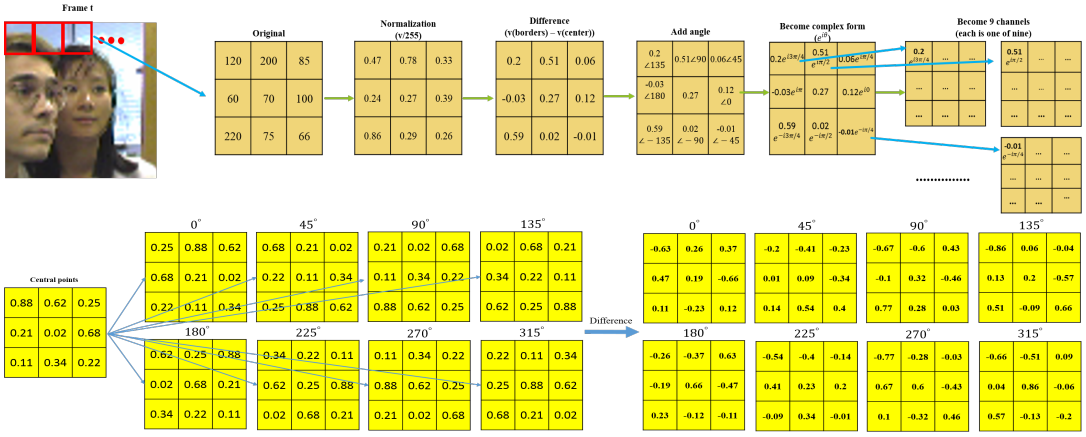

Fig. 2. Comp-LOP feature extraction. Top: Local features. Bottom: Holistic features. **v** means the pixel value.

search region s. Then, we compute the entropy of the target $E(t)$ and the search region $E(s)$ where $E(\cdot)$ is the entropy and t is the target. We obtain the entropy ratio R between $E(t)$ and $E(s)$ as follows:

$$R(k) = E(t)/E(s) \tag{5}$$

where k is the number of calculation times and we update s after each calculation as follows:

$$s = s - 0.2 \tag{6}$$

We repeat (5) and (6) until s becomes 1.2 times of t, i.e. n times. Then, we find a ratio for balancing as follows:

$$\hat{k} = \arg\min_{\mathbf{k}} |R(k) - \frac{1}{n}\sum_{i=1}^{n} R(i)| \tag{7}$$

where $|\cdot|$ stands for the absolute value. Finally, we get the most appropriate search region of s by finding $\hat{k}$ as follows:

$$s = \mu * t - (\hat{k} - 1) * 0.2 \tag{8}$$

where μ is the ratio between the original search region size and the target size.

3.3. Scale Update

By (4), we get the target position with the maximal response $\hat{y}$ in the current frame. However, the scale of the target changes frequently, which may influence the tracking performance. Thus, we update the scale parameter σ in (3) as follows:

$$\theta'_t = (\frac{Y(t) + \theta_t}{Y(t-1) + \theta_{t-1}})^{0.5} \tag{9}$$

$$\bar{\theta}_t = \frac{1}{m}\sum_{i=1}^{m} \theta'_{t-i} \tag{10}$$

$$\theta_{t+1} = (1-\alpha)\theta_t + \alpha\bar{\theta}_t \tag{11}$$

$$\sigma_{t+1} = \sigma_t \theta_t \tag{12}$$

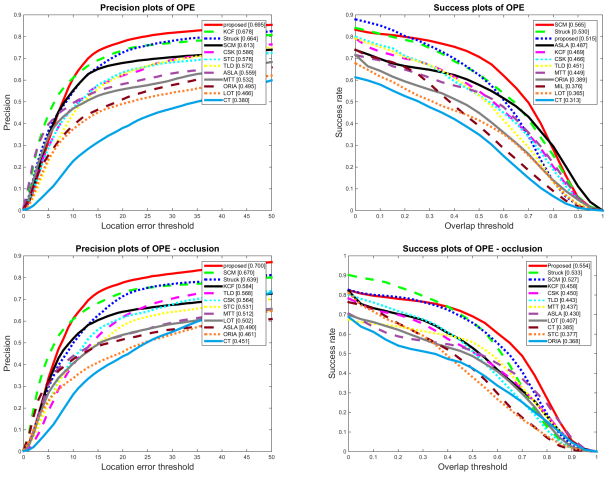

Fig. 3. DP and OS plots by one-pass evaluation (OPE). The left column shows the average DP score at 20 pixels in each tracker, while the right column reports the average area under curve (AUC) score in each tracker.

where $Y(\cdot)$ is the maximum response at a certain frame, and θ'_t is the estimated scale in the frame t. Because the scale of target in two consecutive frames are very similar, it changes little. Thus, we add θ in the numerator and denominator, where θ is the real scale of targets. Also, we use $\bar{\theta}_t$ from m consecutive frames, i.e. the average of the estimated scales, to update the scale. Thus, we obtain the target scale θ_{t+1} of $t+1$ frame and σ by Eqs. (11) and (12), where $\alpha > 0$ is a fixed parameter.

4. EXPERIMENTAL RESULTS

We perform experiments on a large benchmark data set [13]. All the tracking methods are evaluated by three metrics:

Table 1. Performance comparison between the proposed method and other 11 state-of-the-art trackers

Metrics	Proposed	CSK	Struck	MTT	CT	KCF	STC	ORIA	TLD	ASLA	LOT	SCM
DP(%)	**69.5**	58.6	66.4	53.2	38.0	*67.8*	57.6	49.5	57.2	55.9	46.6	61.3
OS(%)	51.5	46.6	*53.0*	44.9	31.3	46.9	40.4	38.9	45.1	48.7	46.6	**56.5**
Speed(fps)	67.6	*150*	12.2	2.1	36.5	86.3	**286**	8.2	21.5	6.4	0.6	0.82

The bold numbers indicate the best performance, the italic ones indicate the second performance, and the underline ones indicate the third performance.

Table 2. Performance comparison of the proposed method and other 11 state-of-the-art trackers under occlusion

Metrics	Proposed	CSK	Struck	MTT	CT	KCF	STC	ORIA	TLD	ASLA	LOT	SCM
DP(%)	**70.0**	56.4	63.9	51.2	45.1	58.4	53.1	46.1	56.8	49.0	50.2	*67.0*
OS(%)	**55.4**	45.0	*53.3*	43.7	38.5	45.8	37.7	36.8	44.3	43.0	40.7	52.7

The bold numbers indicate the best performance, the italic ones indicate the second performance, and the underline ones indicate the third performance.

(i) Distance precision (DP), which shows the percentage of frames whose estimated location is within the given threshold of the ground truth; (ii) Overlap success rate (OS), which is defined as the percentage of frames where the bounding box overlap surpasses a threshold; (iii) Speed, which indicates whether the tracker is real-time or not. We set the threshold=20 pixels in DP and the threshold=0.5 in OS. In (1), we use a gaussian kernel function in Φ. We set the regularization parameter λ in (1) to 10^{-4}. We set the initial size of the search window to 4 times of the target size. The initial scale σ of (3) is $\sigma = \frac{1}{30}\sqrt[2]{wh}$, where w and h are the width and height of the target, respectively. We set the learning rate $\alpha = 0.25$ and $m = 5$ in (9) $\sim$ (12). We make the parameters fixed for all the sequences. The proposed method is implemented on a PC with an Intel I7-6700 3.40 GHz CPU and 8 GB RAM using Matlab.

We evaluate the performance of the proposed method on the benchmark dataset [13] in comparison with other 11 state-of-the-art trackers: CSK [7], STC [4], TLD [3], Struck [12], SCM [14], CT [6], KCF [8], LOT [2], ORIA [15], MTT [16], ASLA [17]. Fig. 3 shows the tracking performance in terms of one-pass evaluation (OPE). Table 1 shows performance comparison between the proposed method and other 11 state-of-the-art trackers. Among the trackers, KCF has the second performance of 67.8% in DP, and SCM has the best performance of 56.5% in OS. The proposed method achieves the best performance of 69.5% in DP and the third performance of 51.5% in OS. The speed of our method is 67.6 frame/sec (fps), i.e. real-time. Moreover, we further analyze their tracking performance on occlusion in Table 2. From Table 2, it can be observed that the proposed method achieves the best performance of 70.0% in DP and of 55.4% in OS. SCM achieves the second performance of 67.0% in DP, while Struck achieves the second performance of 53.3% in OS. That is, the experimental results indicate that the proposed method is more robust against the occlusion. Fig. 4 shows tracking results by some tracking methods on 6 challenging test sequences: CT [6], KCF [8], STC [4], CSK [7], and the proposed method. Although the target objects in *Girl*, *Tiger2*, *Jogging2*, *Coke* sequences are occluded at times, the proposed method still tracks them, but the others fail to track them or track them inaccurately. Besides, the proposed method is robust to the illumination and pose variation as shown in *Skating* and *Shaking* sequences. Furthermore, as shown in *Girl* and *Tiger2* sequences, the proposed method outperforms the other trackers in rotation and scale variation.

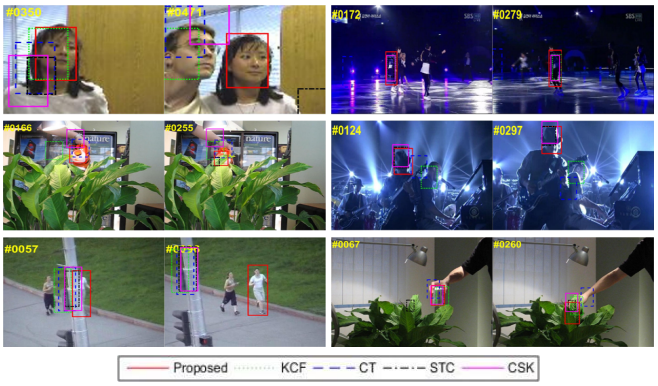

Fig. 4. Tracking results by CT [6], KCF [8], STC [4], CSK [7], and the proposed method. Left-to-right and top-to-down: *Girl*, *Skating*, *Tiger2*, *Shaking*, *Jogging2*, and *Coke*.

5. CONCLUSIONS

In this paper, we have proposed Comp-LOP for object tracking. We have introduced complex form into object tracking to consider the spatiotemporal information between pixels, which is very effective in the occlusion problem. We have utilized the entropy to compute an appropriate search region. Furthermore, we have provided a scale update scheme to make the tracker scale-invariant. Experimental results show that the proposed method outperforms state-of-the-art trackers on large benchmark data sets (DP: 69.5%, OS: 51.5%). Its processing speed is 67.6 fps, i.e. real-time.

6. REFERENCES

[1] Alper Yilmaz, Omar Javed, and Mubarak Shah, "Object tracking: A survey," *Acm Computing Surveys*, vol. 38, no. 4, pp. 81–93, 2006.

[2] Shaul Oron, Aharon Bar-Hillel, Dan Levi, and Shai Avidan, "Locally orderless tracking," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 213–228, 2015.

[3] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[4] Kaihua Zhang, Lei Zhang, Qingshan Liu, David Zhang, and Ming-Hsuan Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proceedings of European Conference on Computer Vision*. Springer, 2014, pp. 127–141.

[5] Hanxi Li, Yi Li, and Fatih Porikli, "Deeptrack: Learning discriminative feature representations online for robust visual tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1834–1848, 2016.

[6] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang, "Real-time compressive tracking," in *European Conference on Computer Vision*. Springer, 2012, pp. 864–877.

[7] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of European conference on computer vision*. Springer, 2012, pp. 702–715.

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

[9] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1090–1097.

[10] Ting Liu, Gang Wang, and Qingxiong Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4902–4912.

[11] Yao Sui, Ziming Zhang, Guanghui Wang, Yafei Tang, and Li Zhang, "Real-time visual tracking: Promoting the robustness of correlation filter learning," in *Proceedings of European Conference on Computer Vision*. Springer, 2016, pp. 662–678.

[12] Sam Hare, Amir Saffari, and Philip HS Torr, "Struck: Structured output tracking with kernels," in *Proceedings of the IEEE Conference on International Conference on Computer Vision*. IEEE, 2011, pp. 263–270.

[13] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.

[14] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang, "Robust object tracking via sparsity-based collaborative model," in *Proceedings of the IEEE Conference on Computer vision and pattern recognition*. IEEE, 2012, pp. 1838–1845.

[15] Yi Wu, Bin Shen, and Haibin Ling, "Online robust image alignment via iterative convex optimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1808–1814.

[16] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra 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.

[17] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of the IEEE Conference on Computer vision and pattern recognition*. IEEE, 2012, pp. 1822–1829.