Deep Convolutional Features for Correlation Filter Based Tracking with Parallel Network

Jinglin Zhou, Rong Wang(✉), and Jianwei Ding

People's Public Security, University of China, Beijing, China
henlouser@163.com, dbdxwangrong@163.com

Abstract. Visual tracking has made great progress in either efficiency or accuracy, but still remain imperfections in accurately tracking on the premise of real time. In this paper, we propose a parallel network to integrate two trackers for real-time and high accuracy tracking. In our tracking framework, both trackers are based on correlation filters running in parallel, with one using hand-crafted features (tracker A) for efficiency and another using deep convolutional features (tracker B) for accuracy. And the tracking results are under supervision by a novel criterion. Furthermore, the sample models trained for correlation filter are optimized by controlling sampling frequency. For evaluation, our tracker is experimented on the datasets OTB2013 and OTB2015, demonstrating a higher accuracy than the state-of-the-art trackers on the premise of real time, especially in the situation of object deformation and occlusion.

Keywords: Tracking · Convolutional feature · Correlation filter
Parallel network

1 Introduction

Object visual tracking is the fundamental work in the field of computer vision. With the broad application prospects in high-level visual tasks, such as automatic driving [1] and scene understanding [2], visual tracking has attracted great attentions from researchers [3, 4]. Whereas, suffering from the interference introduced by environment variation, generic visual tracking is still one of the most challenge research in computer vision. A robust tracking algorithm with high accuracy and efficiency becomes a research hotspot nowadays [5].

Since the deep convolutional neural networks have made a significant breakthrough in the field of object recognition [6], researchers have begun to apply the deep neural network architecture to visual tracking. The convolutional neural network based tracking algorithms have demonstrated great advantages in tracking accuracy [7]. Based on deep convolutional neural networks, trackers are pre-trained by relative sequences for a robust end to end tracker [8]. Another application is exacting deep convolutional features for Siamese network [9], aiming at an online learning tracker. The deep convolutional network has made some progress in accuracy, it performs poorly in terms of computation speed [10].

Y. Wang et al. (Eds.): IGTA 2018, CCIS 875, pp. 461–470, 2018.
https://doi.org/10.1007/978-981-13-1702-6_46

The tracking algorithms based on the correlation filter have improve the computing efficiency to a certain extent. The first correlation filter based tracking algorithm MOSSE [11] had caused a sensation for its excitement efficiency. Since then, the KCF [12] tracker applies the kernel function and cyclic matrix to the correlation filter, which is the further improvement of this kind algorithm. In the subsequent SRDCF [13] tracker, the scale adaption is introduced for the problem of object transforming scales continuously. Nevertheless, the correlation filter based trackers have an inherent limitation that they resort to low-level hand-crafted features, are easy to affected by the interference when target is occluded or blurred.

In this paper, deep convolutional neural feathers and online sample model optimization mechanism are introduced for a correlation filter based tracker, aiming at both real-time and high accuracy tracking. Firstly, features are exacted from the first frame and integrated to the sample model for correlation filters. Secondly, based on a parallel implementation architecture, two tracker which based on correlation filter but with different features do tracking alternately and have complementary advantages. The tracker with hand-crafted features executed tracking in most frames under supervise and the other tracker with deep convolutional features would be activated to rectify the tracking results when the results from the former are considered to unreliable. Finally, the tracking results are selected to update the sample model under supervise, for the online training of correlation filters which would be used for the next frame.

The proposed tracker is evaluated on the popular datasets OTB2013 and OTB2015, comparing to 8 state-of-the-art trackers. The experiments show that our tracker have advantages in the situation that the target is blurred and under deformation, and could keep a high accuracy in real-time tracking.

2 Method

2.1 Deep Convolutional Neural Network

Convolutional neural network is a classical architecture of deep learning that inspired from visual mechanism of biology, having a great advantage in exacting robust features which are applicable for object rotation, deformation and so on. Furthermore, without designing, the convolutional neural network could achieve different dimensional features.

According to above, we employ the Very Deep Convolutional Network [14] (which is notated as VGG Net in later parts of this paper) as the convolutional neural network we exacted features from. There are 19 weight layers in the VGG Net we employed, as shown in Fig. 1.

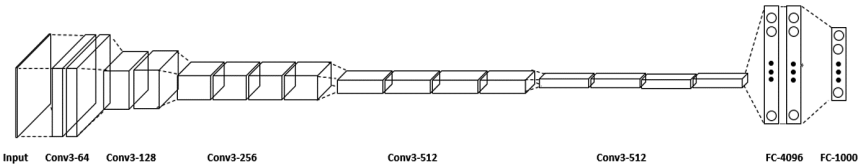


Fig. 1. The structure of the VGG Net, including 19 weight layers. There are 16 convolutional layers, 3 fully-connected layers and 5 max-pooling layers.

To exact features, we pre-trained the VGG Net on the ImageNet dataset. Experiments show that the deep layers have achieve richer semantic information and less detail texture, whereas the shallow layers achieved the reverse. The visualization of the features exacted from different layers have been shown in Fig. 2.

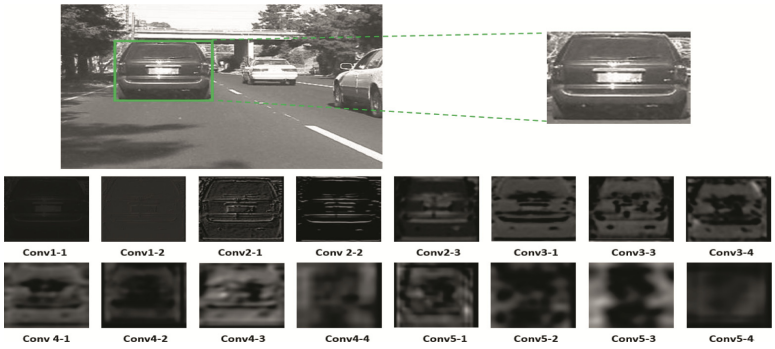


Fig. 2. The visualization of features exacted from every layer in VGG Net. With the deepening of the layer, we could obtain more semantic information and less details. For a complete expression of samples, we employ features from both shallow and deep layers in this paper.

2.2 Multi-features Integration

To make the best of the advantages in diverse features, we select features from multiple layers. According to experiments, we employ the feathers exacted from conv1-1, conv2-2 and conv5-2. And we also employ hand-craft feathers including HOG and color names features to improve the efficiency of the algorithm. The convolutional features and the hand-craft feathers work in a parallel framework, and more detail discussion we would made in Sect. 3.1.

As there are more than one features in each streams, the integration of multiple features should be taken into account. The traditional method is utilizing features by integrating all feature maps, with massive computation.

The other method proposed by [4] performs more efficient. Their method reduce computation by transforming different features into a continuous spatial domain. According to this method, we defines $x_1, x_2, \ldots, x_i$ that represent the training samples for correlation filter from image patches. For each samples, there are n feature channels

for sample x_i that we notate as $x_i^1, x_i^2, \ldots, x_i^n$. And the integration feature function is $J\{x\}$. Assuming that there are m (m < n) feathers have made the most contribution, we could reduce a deal of computation with very little cost by discarding some less-contribute feature channels. Here we introduce an M × N matrix P, then we could represents the integration feature $J\{x\}$ approximately as

$$J\{x\} = P^T J_n\{x^n\}. \tag{1}$$

2.3 Correlation Filter

The correlation Filter is usually applied to evaluate the correlation between two signals in communication filed. Extending to visual tracking problem, it is a realistic solution that considering object tracking as searching the most correlation region between two frames. The correlation filter based tracking algorithm model the samples and transform the signal from time domain into frequency domain by Fast Fourier Transform (FFT). Benefiting from the high efficiency in Fourier domain, the correlation filter based tracking algorithm enjoy a high computing speed. Here, we denote the input image window as w and the trained filter as f. After Fourier transformation, we would obtain the function in frequency domain as

$$\mathrm{W} = \mathcal{F}(w). \tag{2}$$

$$\mathrm{F} = \mathcal{F}(f). \tag{3}$$

The correlation takes the form

$$\mathrm{G} = \mathrm{W} \odot F^*. \tag{4}$$

Where the ⊙ indicates element-wise multiplication and the * indicates the complex conjugate. The location corresponding to the maximum value of G represents the new position in the current frame. Base on deduction in Sects. 3.2 and 3.3, there is

$$\mathrm{G} = \mathrm{W} \odot F^* = f * P^T J_n\{x^n\}. \tag{5}$$

According to above, the tracking task have been transformed to finding the position that the maximum correlation responding to. Furthermore, the filter would update online based on the sample obtained from the new frame.

3 Our Tracking Framework

3.1 Architecture

Our tracking framework consist of two trackers. Both of the trackers are based on the correlation filter, with one of them using deep convolutional features (Tracker A) and the other one using hand-craft features (Tracker B). Tracker B has advantage in computation efficiency, along with Tracker A represents more reliable on tracking accuracy.

Each tracker performs its own functions in parallel, aiming at both high accuracy and real-time tracking. Here we employ the SRDCF [13] tracker as our baseline tracker.

As shown in Fig. 3, tracker A executes tracking task and evaluates whether its result is reliable in every frame. The tracking task would continue if the result was considered to be confidence. Otherwise, a request would be sent to tracker B, asking for a more reliable result from tracker B who is based on deep convolutional feathers. The tracker B would not execute tracking unless received request. At the rest of time, tracker B only updates the sample model based on the tracking results from tracker A.

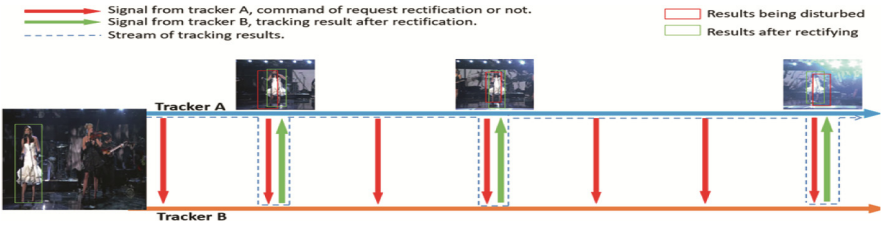


Fig. 3. The pipeline of our tracking framework. The two trackers work in parallel under supervision. The tracker A do tracking in most frames and request of rectifying would be sent to tracker B if the results from tracker A is considered to be unreliable.

3.2 Confidence Embedding

According to the theory of the correlation filter based trackers, the location of maximum correlation responding would be considered to be the position of target. But with the interference such an illumination variation or occlusion, the location of maximum correlation responding might not be the most appropriate position to the target. What is more, there is a serious risk that the tracking results occurred to drift in serval frames letter when the mistaken is accumulated without rectifying promptly, as shown in Fig. 4.

Fig. 4. A comparison of tracking with rectifying and without rectifying. The tracking results is drifting when errors from interference were not rectified promptly. The response map on the right demonstrates the correlation response in frame 157 where errors started accumulate, suffering a low level of PD.

In order to detect the problematic tracking results and rectify it timely, we employ two criteria to judge whether tracking results are reliable or not. One of them is the value of G which has discussed in Sect. 2.3, the other one is a measure we creatively proposed.

By repeating experiment, the problematic tracking results always happened to the situation that there exist several regions similar to the target in appearance nearby. Reflected on the convolutional correlation response maps, there would be several peaks distributing on the map. Nevertheless, under normal conditions there is only one peak on the correlation response map that the value this peak responding to is much higher than the other value is. In view of this, we could evaluate the reliability of the tracking results by analyzing the distribution of the correlation response scores on the map.

In every frame, there are 2809 (53 × 53) response scores being calculated, which agree with the Gaussian distribution in numerical value. Here we denote that the response scores as G_{sc}, the expectation of G_{sc} as μ_{sc} and the standard deviation as σ_{sc}.

$$G_{sc} \sim N\left(\mu_{sc}, \sigma_{sc}^2\right) \tag{6}$$

According to the theory of statistics, the standard deviation is a measure of the distribution. A higher value of the standard deviation indicates there is only one peak in the map instead of several. The standard deviation σ_{sc} could be defined as

$$\sigma_{sc} = \sqrt{\sum \left(G_{sc} - \mu_{sc}\right)^2}. \tag{7}$$

Here we obtain a new measure to evaluate the distribution of the peaks on the map as Peaks Distribution (PD)

$$\mathrm{PD} = \sigma_{sc}. \tag{8}$$

On the basis of above, we could consider a high value of PD indicates that the maximum response is significant and the noise is small. In other words, the tracking results are reliable in the situation of the value of PD is high enough.

Here we propose the conditions which must be satisfied in the situation of tracking results considered to be reliable: $\mathrm{G} > T_1$ *and* $PD > T_2$. Where T_1 and T_2 is pre-setting before tracking and keep on updating in the online learning.

As the tracking results in tracker A in the architecture we have discussed in Sect. 4.1 are considered to be unreliable, a request signal would be sent to tracker B asking for rectifying.

3.3 Online Optimizing

Here, we propose a strategy to optimize the sample model for the correlation filter training. Most correlation filter based trackers update the sample model in every frame, without considering whether the updates are necessary. Whereas, the unreliable updating would bring about errors and frequent updating would lead to overfitting.

Picking up the appropriate samples for correlation filter training instead of every exacted samples could not only improve the accuracy rate of tracking, but also reduce the computational complexity effectively, especially for the tracker which is based on deep convolutional features.

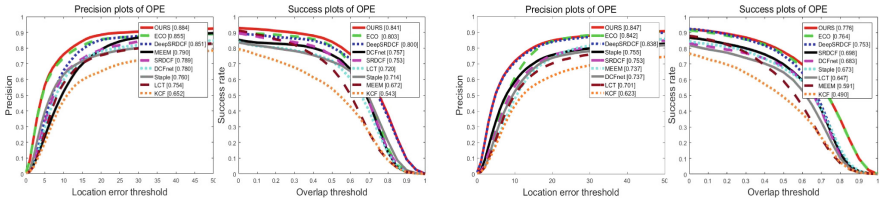
Our strategy of the sample model optimization is updating model under online learning. For avoiding the overfitting, we distribute the samples in groups. The successive similar samples are divided into one group (as shown in Fig. 5) and the frequent in different group is unequable. Here we introduce a criteria learning rate ζ to indicate the frequency of sampling. The more samples accumulated in one group, the lower the learning rate ζ is in this group. To ensure the reliability of samples, the correlation filter would simply be trained when results are considered to be confidence. Here we employ the measure PD we have discussed in Sect. 3.1.

$$\zeta = \begin{cases} 0 \left(PD < \mathrm{T}_3\right) \\ \dfrac{k}{PD} (k > 0, PD > \mathrm{T}_3) \end{cases} \tag{9}$$

Fig. 5. An example of grouping the samples. The successive similar samples are divided into group, for reducing the computation and avoiding overfitting.

4 Experiments

We extensively evaluate our tracker on the most popular visual tracker benchmark nowadays, the Object Tracking Benchmark (OTB) [16]. The OTB consists of two datasets, OTB2013 and OTB2015, which consist of 50 and 100 sequences respectively. For the purpose of comparison, we employ 8 state-of-the-art trackers, including ECO [15], DeepSRDCF [13], SRDCF [13], Staple [17], LCT [18], DCFNet [8], MEEM [19] and KCF [12], as shown in Fig. 6.



(a) Comparisons on OTB 2013 (b) Comparisons on OTB 2015

Fig. 6. Comparisons with state-of-the-art tracking trackers on OTB2013 and OTB2015 with the DPR and OSR.

4.1 Implementation Details

Our tracker is implemented in a numerical computing environment, Matrix Laboratory (Matlab) in version 2016b. The experiments is validated on a machine equipped with an Intel Core i7 running at 2.50 GHz with 16 GB memory.

4.2 Experiments on OTB2013 and OTB2015

The Object Tracking Benchmark (OTB) is one of the most popular tracking benchmark in the word. The OTB2013 and OTB2015 are two datasets published in 2013 and 2015, which consist of 50 and 100 sequences, respectively.

Attribute-Based Evaluation. For each sequences in OTB, there are 11 attribution is annotated on it, including illumination variation, scale variation, motion blur, in-plane rotation, occlusion, fast motion, deformation, out-of-plane rotation, out of view, low resolution and background clutter. We further analyze our tracker under different attributes in OTB2015. In terms of DPR and OSR, our tracker obtain best results under 9 out of 11 attributes. Owing to the parallel network and rectifying mechanism, our tracker achieves higher accuracy and success rate than others. Nevertheless, the proposed tracker still has difficulties in low resolution and out-of-view, showing that there is room for online learning of threshold and model optimization.

Qualitative Evaluation. Figure 7 demonstrates qualitative comparisons of our tracker with eight state-of-the-art trackers on four sequences selected from OTB2013. Compared with the correlation filter based tracker, our tracker performs more reliable in sequences with deformation and rotation. The deep convolutional feature based trackers could deal with these cases, but failed in sequences with occlusion. Our tracker performs the best in these sequence, benefiting from the parallel architecture.

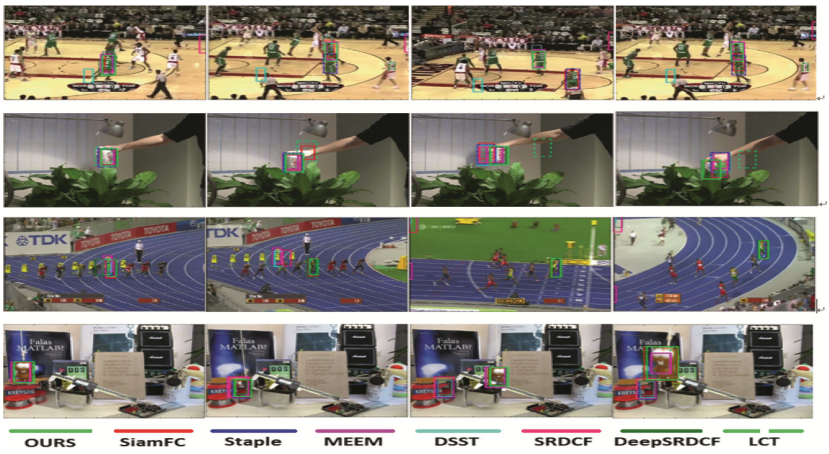


Fig. 7. Qualitative evaluation of the proposed tracker and other eight state-of-the-art trackers on four sequence in OTB2015 (from top to bottom: *Basketball*, *Coke*, *Bolt*, and *Sylvester*).

5 Conclusion

In this paper, we propose a novel real-time object tracking method by integrating deep convolutional features with correlation filter using a parallel network. A strategy is proposed to verify the current tracking results according to convolution response and rectify the results by the deep features based component. Making the best of high efficiency from correlation filter and reliability from deep features. Furthermore, we construct an online sample model optimizing strategy to reduce computation toward more efficiency tracking. The encouraging results are demonstrated in experiments performed on OTB2013 and OTB2015, achieving a state-of-the-art performance both on accuracy and on speed. Further work would involve the improvement of the rectified performance by introducing a more reliable tracker for rectifying.

Acknowledgments. This work is supported by National Key Research and Development Plan under Grant No. 2016YFC0801005. This work is supported by the National Natural Science Foundation of China under Grant No. 61503388.

References

1. Pohlen, T., Hermans, A., Mathias, M., Leibe, B.: Full-resolution residual networks for semantic segmentation in street scenes. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3309–3318. IEEE Press, Hawaii (2017)
2. Bagautdinov, T., et al.: Social scene understanding: end-to-end multi-person action localization and collective activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3425–3434. IEEE Press, Hawaii (2017)
3. Galoogahi, H.K., et al.: Learning background-aware correlation filters for visual tracking. In: IEEE International Conference on Computer Vision, Venice, pp. 1135–1143 (2017)
4. Danelljan, M., Robinson, A., Shahbaz Khan, F., Felsberg, M.: Beyond correlation filters: learning continuous convolution operators for visual tracking. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 472–488. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_29
5. Fan, H., Ling, H.: Parallel tracking and verifying: a framework for real-time and high accuracy visual tracking. In: IEEE International Conference on Computer Vision, Venice, pp. 5486–5494 (2017)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
7. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE Press, Las Vegas (2016)
8. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.H.S.: End-to-end representation learning for correlation filter based tracking. In: IEEE International Conference on Computer Vision, Venice, pp. 5000–5008 (2017)
9. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 850–865. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_56

10. Kang, K., et al.: Object detection from video tubelets with convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 817–825. IEEE Press, Las Vegas (2016)
11. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2544–2550. IEEE Press, San Francisco (2010)
12. Henriques, J.F., et al.: High-speed tracking with kernelized correlation filter. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, pp. 583–596 (2015)
13. Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: IEEE International Conference on Computer Vision, pp. 4310–4318 (2016)
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Comput. Sci. (2014)
15. Danelljan, M., et al.: ECO: efficient convolution operators for tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6931–6939. IEEE Press, Hawaii (2017)
16. Wu, Y., Lim, J., Yang, M.-H.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
17. Bertinetto, L., et al.: Staple: complementary learners for real-time tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1401–1409. IEEE Press, Las Vegas (2016)
18. Ma, C., Yang, X., et al.: Long-term correlation tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5388–5396. IEEE Press (2016)
19. Zhang, J., Ma, S., Sclaroff, S.: MEEM: robust tracking via multiple experts using entropy minimization. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 188–203. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_13