Accurate Scale-Variable Tracking

Xinyou Li, Wenjing Kang(✉), and Gongliang Liu

School of Information and Electrical Engineering,
Harbin Institute of Technology, Weihai, China
13115416536@163.com, {kwjqq,liugl}@hit.edu.cn

Abstract. In recent years, several correlation tracking algorithms have been proposed exploiting hierarchical features from deep convolutional neural networks. However, most of these methods focus on utilizing the CNN features for target location and neglect the changes of target scale, which may import error to the model and lead to drifting. In this paper, we propose a novel scale-variable tracking algorithm based on hierarchical CNN features, which learns correlation filters to locate the target and constructs a target pyramid for scale estimation. To evaluate the tracking algorithm, extensive experiments are conducted on a benchmark with 100 video sequences, which demonstrate features exploited from different CNN layers are well fit to estimate the object scale. The evaluation results show that our tracker outperforms the state-of-the-art methods by a huge margin (+14.6% mean OS rate and +14.3% mean DP rate).

Keywords: Correlation tracking · Scale estimation · CNN features

1 Introduction

Object tracking is a fundamental problem in computer vision with several applications such as video surveillance, medical diagnosis and human-computer interactions. However, the interference factors like illumination, occlusion, scale variations and abrupt motion make visual tracking still a challenging problem.

Many exiting tracking algorithms utilize hand-crafted features as target descriptors [1, 2], but recent years deep Convolutional Neural Networks (CNNs) features have demonstrated great success on object presentation. Thus recent algorithms utilize CNNs features to train correlation filters to predict target position [3, 4]. However, these algorithms do not take object scale variation into account and the error would stimulate when the target undergoes scale changes, which would eventually lead to drifting or tracking failure. This issue is the well-known stability-plasticity dilemma. In this paper, we effectively alleviate this dilemma by integrating target location and scale estimation. We generate a translation template using correlation filters for target location and scale models to construct a target pyramid for scale estimation. The scale model utilizes the predicted target position to search for the optimal scale, and the estimated target size in return helps to generate a more stable translation model for target location.

Except for scale variation, there are other video attributes would affect tracking performance. However, most of the existing methods using HOG features to construct the target pyramid, while CNN features are prevailing in high-level visual recognition

X. Gu et al. (Eds.): MLICOM 2017, Part I, LNICST 226, pp. 40–48, 2018.
https://doi.org/10.1007/978-3-319-73564-1_5

problems because of the robustness against attributes like motion blur or illumination variation. We also find that hierarchical CNN features retain semantic information and spatial details, which are both needed in modeling the target. With these observations, we propose to utilize hierarchical CNN features to build the target pyramid. Moreover, we conceive a new approach to extract scale features in the target pyramid by using a CNN to scan the image computing a large feature map, which effectively reduce computational load and demonstrate great success.

We make the following three contributions. First, we alleviate the stability-plasticity dilemma by integrating target location with object scale estimation. A target pyramid is constructed centered around predicted target location to determine the object scale, and the translation template is updated considering estimated object size to locate the target position. The integrating tracking strategy effectively reduces tracking drifts and remarkably improves the performance in videos with scale variation. Second, we innovatively propose to utilize hierarchical CNN features to generate the target pyramid. We extract every scale features in target pyramid with a scan from the CNN. Features from different layers of a CNN retain spatial details and sematic information, which are both helpful to encode scale models robust against motion blur and illumination variation. Third, we conduct extensive experiments on a large-scale benchmark dataset with 100 video sequences [5]. The tracking results demonstrate the effectiveness of our proposed accurate scale-variable tracking algorithm (AST).

2 Related Work

Heriques et al. first exploit circulant structure of training samples and propose to transfer correlation filters into the Fourier domain with CSK method, which reaches a speed of about 250 frames per second [6]. Furthermore, the KCF tracking algorithm uses HOG features other than illumination intensity features and improves the performance of CSK [7]. In [8], Bolme et al. learn a minimum output sum of squared error filter on gray-scale images, using intensity features to represent the object.

Recent years deep CNNs have improved state-of-the-art performance in many computer vision tasks, and some researchers attempt to explore the usage of CNNs in visual tracking. Ma et al. develop a correlation tracker based on hierarchical features from a deep CNN. Due to its coarse-to-fine translation estimation strategy, the HCF tracker can locate the target precisely. Qi et al. combine weak CNNs based trackers into a single stronger tracker [4]. However, these trackers do not take target scale changes into account and cannot perform well when target undergoes scale variation.

For scale estimation, Danelljan et al. propose to construct target pyramid around the object, and their fast scale tracking algorithm with HOG features performs well in overlap success rate with a considerable speed [9]. Ma et al. learns a multi-level correlation filters to estimate target scale, but they do not use estimated scale to improve positioning accuracy [10]. In this paper, we exploit hierarchical features for different CNN layers to build a target pyramid and train two models separately for predicting position and scale estimation. We conduct extensive experiments on large-scale benchmark datasets, and the results demonstrate the effectiveness of our algorithm, especially when tracking sequences with scale variation, motion blur and deformation.

3 Proposed Algorithm

3.1 Correlation Tracking

Let $x \in \mathbb{R}^{M \times N}$ denotes feature vector of size $(M \times N)$. Each shifted sample $x_{m,n}$, $(m,n) \in \{0,1,\cdots,M-1\} \times \{0,1,\cdots,N-1\}$ has a Gaussian Function label of $y(m,n) = exp\left(-\left((m-M/2)^2 + (n-N/2)^2\right)/2\sigma^2\right)$, where σ is the kernel width. A correlation filter w is generated by solving following minimization problem:

$$w = \arg\min_{w} \sum_{m,n} \left\| w \cdot \varphi\left(x_{m,n}\right) - y(m,n)^2 \right\| + \lambda \|w\|_2^2 \tag{1}$$

where $\varphi\left(x_{m,n}\right)$ denotes the mapping to a kernel and λ is a regularization parameter.

Henriques et al. [6] exploit the circulant structure of training samples $x_{m,n}$ and transform the minimization problem in (2) to compute the coefficient α in $w = \sum_{m,n} \alpha(m,n) \cdot \varphi\left(x_{m,n}\right)$. And α can be computed in frequency domain:

$$A = \mathcal{F}(\alpha) = \frac{\mathcal{F}(y)}{\mathcal{F}(\varphi(x) \cdot \varphi(x)) + \lambda} \tag{2}$$

$\mathcal{F}(\cdot)$ indicates the Fourier transform. The position of target in new frame is determined by searching for the location of the maximal value of correlation response map.

3.2 Scale Estimation

According to [9], let N be the number of scales with a scale factor of a. For every $n \in \{-(N-1)/2, \cdots, (N-1)/2\}$ we extract image patch I_n of size $\mathrm{S}_n = \alpha^n \cdot [h,w]$ centered around the target, where $[h,w]$ is the target size in previous frame. For each image patch I_n we extract CNNs features then compute response map p_n and find the maximal value of each p_n. The optimal target scale for currant frame is determined by:

$$\mathrm{S} = \mathrm{S}_n = arg\max_{n}(p_n) \tag{3}$$

Note that we train two correlation filter R_t and R_s separately for target location and scale estimation. And R_t incorporates both target and surrounding context information because this information can effectively discriminate the target from background [11]. In contrast, R_s only depend on the target size for robust scale estimation.

3.3 Deep CNN Features

Several CNN models, such as AlexNet, R-CNN, CaffeNet and VGG-Net have been designed and demonstrate great success in large-scale image classification and object recognition tasks. According to Ma et al. [3], the features learned from latter CNNs layers encode more semantic information and earlier layers retain higher spatial resolution, which are both needed in tasks of target location and scale estimation.

Therefore, we propose to utilize hierarchical features from VGG-NET-19 [12] for translation template and scale models.

According to traditional method, we must first crop out windows of every scale in target pyramid and then obtain hierarchical features using a CNN. It means that we need to repeat extracting CNN features every frame. Since the process of forward propagation of a CNN requires large amount computing time, and these scale features retain many repeating information. Based on these observations, we propose to use CNN to scan the whole image and then gain all scale features at once. We first use target pyramid to compute the size of searching window adjusted by previous target size, then we crop out a window from the image and gain its CNN feature maps, finally we extract features of every scale in target pyramid from the large feature maps at once.

3.4 Model Update

In our proposed algorithm, we train two models R_t and R_s separately for target location and scale estimation. Since the target appearance would change throughout a sequence, we update the models every frame by a learning rate η:

$$\tilde{x} = \tilde{x}^{t-1} + \eta\tilde{x}^{t} \tag{4}$$

$$A = A^{t-1} + \eta A^{t} \tag{5}$$

where t is the frame index. Notice that we update R_t and R_s every frame using (4) and (5) with the same learning rate. We predefine a threshold ξ_s and stably update models only when the difference between the response map's maximal value of previous frame and current frame is less than ξ_s.

Algorithm 1 Proposed tracking algorithm: iterate at frame t

Input : Previous target position p_{t-1} and scale s_{t-1},
Output: Estimated target position p_t and scale s_t
Repeat:
Crop out the searching window in frame t according to (p_{t-1}, s_{t-1}) and extract features;
Compute the correlation map yt using R_t to estimate the new position p_t;
Build the target pyramid according to (p_{t-1}, s_{t-1}) and compute the correlation map ys using R_s;
if $|\max(ys_t) - \max(ys_{t-1})| < \xi_s$
Estimate the optimal scale s_t using (3);
else
$s_t = s_{t-1}$;
end
Updated R_t and R_s use (4) and (5);
Until *End of the video sequences*

4 Experiments

4.1 Implemental Details

The main steps of the proposed algorithm are presented in Algorithm 1. We set the regulation parameter of (1) to $\lambda = 10^{-4}$. The number of scale space in target pyramid is set to $S = 21$ with scale factor of 1.03. The learning rate in (4) is set to 0.01. The threshold of updating target scale is set to $\xi_s = 0.1$. We run our implementations in Matlab on HP OMEN 15-AX000 with an Intel I5-6700HQ 2.6 MHz CPU, 4 GB RAM and a GeForce GTX960 GPU card. The GPU card is only used to extract CNN features.

4.2 Comparisons with State-of-the-Art Trackers

We compare our tracker with top 5 state-of-the-art tracking algorithms that are provided in OTB-100 [5]. These algorithms can be divided into three typical categories, (i) correlation tracker (CSK [6], KCF [7]), (ii) tracking by single classifier (MIL [13], Struck [14]), (iii) tracking by multiple online classifier (TLD [15]).

Quantitative evaluation. Figure 1 and Table 1 presents the tracking results on OTB-100. We highlight the best value in Table 1 by bold. Among all 5 trackers, the KCF tracker achieves the highest DP rate of 69.0%, OS rate of 54.6% and CLE of 44.6. And our algorithm outperforms KCF with raises of 14.3% DP rate, 14.6% OS rate and reduction of 21.2 CLE. Note that our tracker runs in a speed of 3.8 frames per second on OTB-100 [5], because the forward propagation process of CNNs has a high computation load.

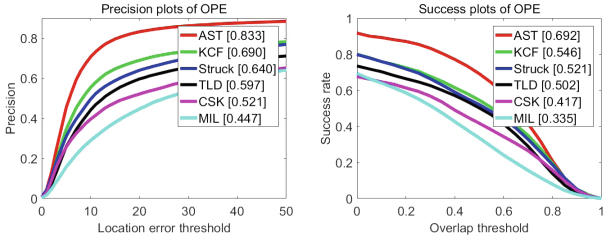

Fig. 1. Distance precision and overlap success plot over OTB-100 using one-pass evaluation (OPE)

Table 1. Comparisons with the state-of-the-art trackers on 100 benchmark sequences

	Ours	CSK [6]	Struck [14]	MIL [13]	TLD [15]	KCF [7]
DP rate (%)	**83.3**	52.1	64.0	44.7	59.7	69.0
OS rate (%)	**69.2**	41.7	52.1	33.5	50.2	54.6
CLE (pixel)	**23.4**	305	47.1	72.1	60.0	44.6
SPEED (FPS)	3.77	**248**	9.84	28.0	23.3	207

Attribute-based evaluation. To further analyze robustness of the proposed algorithm when tracking in various scenes, we evaluate the performance of our algorithm under different video attributes and show the results in Fig. 2. As revealed in Fig. 2, our approach outperforms other methods in all the six tracking challenges. Especially, AST shows its great superiority when tracking the sequences with scale variation, motion blur and illumination variation. The hierarchical features from CNN retain spatial details and semantics, which are both useful for discriminating target from background in fast motion and motion blur sequences. Meanwhile, target pyramid constructed centered around the object effectively predict target scale and stable update strategy helps to generate robust models in videos with scale variation.

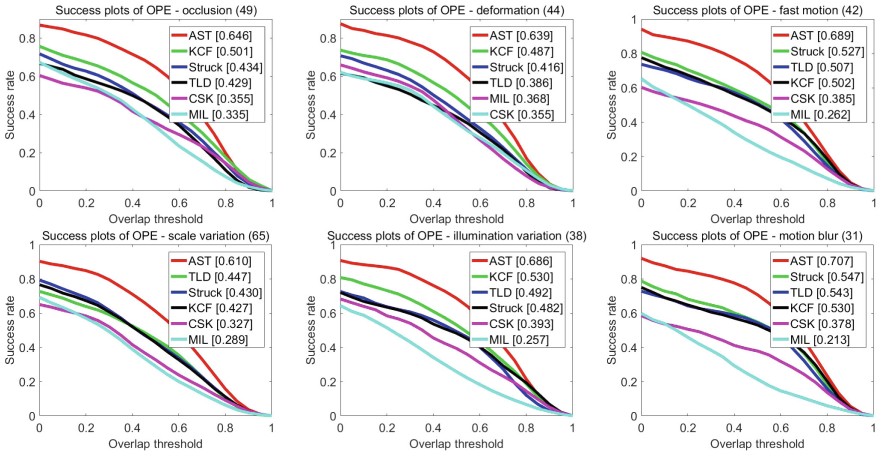

Fig. 2. Overlap success plots over six tracking challenges

Qualitative evaluation. We report tracking results of 5 sequences from 6 trackers in Fig. 3. The CSK tracker learns a kernelized correlation filter for tracking, but the intensity features make the tracker drift when target undergoes rotation, fast motion and partial deformation (Toy, Tiger1, DragonBaby and Skiing). The KCF tracker improve the performance of CSK by using HOG features, but HOG features cannot well discriminate targets in cluttered background and fast motion (DragonBaby and Skiing). The Struck method use structure output to alleviate sample ambiguity, but the HOG features cannot handle large appearance changes and it does not perform well in rotation, deformation and background clutter (Tiger1, DragonBaby and Skiing). The MIL method use multiple instance learning to find positive samples to train the detector. But the insufficient positive samples result in tracking drift caused by fast motion, illumination variation and partial deformation (Toy, Car4, Tiger1, DragonBaby and Skiing). Meanwhile, the TLD method cannot sufficiently exploit semantic information and spatial details, and it prone to drift or even fail to re-detect when comes to fast motion, deformation and partial occlusion (Toy, Tiger1, DragonBaby and Skiing).

There are mainly 3 reasons why the proposed AST tracker performs favorably against the other 5 algorithms. First, we exploit features from different CNN layers to build a target pyramid. The hierarchical features retain both spatial details and semantic information, which are both necessary for target description. Second, we combine correlation tracking with scale changes to alleviate the stability-plasticity dilemma and effectively improve tracking performance. Third, we stably update target scale to gain a robust model and effectively alleviate tracking drifts. As a result, our proposed algorithm effectively handle all the 5 videos.

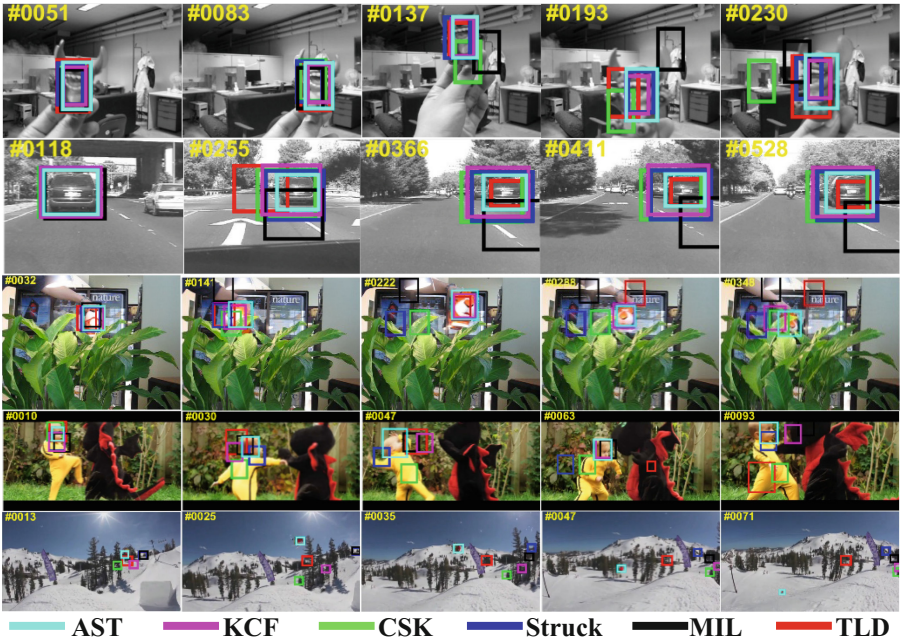

Fig. 3. Qualitative results of AST, KCF [7], MIL [13], CSK [6], Struck [14] and TLD [15] methods on five challenging sequences (Toy, Car4, Tiger1, DragonBaby and Skiing)

4.3 Component Analysis

We further implement three algorithms on benchmark dataset [16] with 50 videos to demonstrate the effectiveness of the proposed algorithm. Except the AST, we generate the ATCNN tracker training correlation filters for target location like AST but remove the target pyramid. Also, we implement the ATHOG tracker training correlation filters and target pyramid both using HOG features. The results are reported in Fig. 4.

As shown in Fig. 4, ATHOG preforms the worst among 3 trackers. Because the target pyramid is constructed centered around the predicted target position, the effectiveness of scale estimation does closely depend on the accuracy of target location. And HOG cannot well describe target appearance in different scenarios. Compared

with AST, the ATCNN tracker neglects target scale changes and trains correlation filters with a fixed- size window. AST uses hierarchical CNN features generating target models with the consideration of scale variation, raising the DP rate to 89.4% and OS rate to 76.9%.

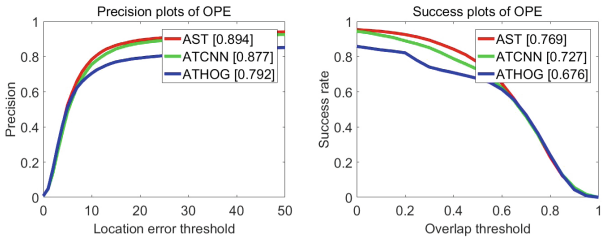

Fig. 4. Comparisons of component effectiveness features

5 Conclusion

In this paper, we propose an effective algorithm for accurate scale-variable tracking. The integrating of target location and scale estimation successfully alleviates the stability-plasticity dilemma caused by scale variation. Meanwhile, scale models trained by hierarchical CNN features remarkably improves the performance in tracking videos with motion blur and illumination variation. Extensive experiment results on a large-scale benchmark demonstrate the great success of the AST tracker.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Grant No. 61501139, 61371100), and the Natural Scientific Research Innovation Foundation in Harbin Institute of Technology (HIT.NSRIF.2013136).

References

1. Jia, X., Lu, H., Yang, M.-H.: Visual tracking via adaptive structural local sparse appearance model. In: CVPR, pp. 1822–1829. IEEE Press, Plantations (2012)
2. 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
3. Ma, C., Huang, J.-B., Yang, X., Yang, M.-H.: Hierarchical convolutional features for visual tracking. In: ICCV, pp. 3074–3082. IEEE Press, Santiago (2015)
4. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., Yang, M.-H.: Hedged deep tracking. In: CVPR, pp. 4303–4311. IEEE Press, Las Vegas (2016)
5. Wu, Y., Lim, J., Yang, M.-H.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)

6. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7575, pp. 702–715. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_50
7. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-Speed tracking with kernelized correlation filters. TPAMI **37**(3), 583–596 (2015)
8. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: CVPR, pp. 2544–2550. IEEE Press, San Francisco (2010)
9. Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: BMVC, pp. 583–596. BMVA Press, Nottingham (2014)
10. Ma, C., Yang, X., Zhang, C., Yang, M.H.: Long-term correlation tracking. In: CVPR, pp. 5388–5396. IEEE Press, Boston (2015)
11. Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.-H.: Fast visual tracking via dense spatio-temporal context learning. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 127–141. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_9
12. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 [cs.CV]
13. Babenko, B., Yang, M.-H., Belongie, S.: Robust object tracking with online multiple instance learning. TPAMI **33**(8), 1619–1632 (2011)
14. Hare, S., Saffari, A., Torr, P.H.S.: Struck: structured output tracking with kernels. In: ICCV, pp. 263–270. IEEE Press, Barcelona (2011)
15. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. TPAMI **34**(7), 1619–1632 (2012)
16. Wu, Y., Lim, J., Yang, M.-H.: Online object tracking: a benchmark. In: CVPR, pp. 2411–2418. IEEE Press, Portland (2013)