High Confidence Tracking with Offline Historical Learning and Online Correlation Filter Updating

Shou-dong HAN\textsuperscript{1,2,*}, Hong-wei WANG\textsuperscript{1} and Xin-xin XIA\textsuperscript{1}

\textsuperscript{1}Key Laboratory of Image Processing and Intelligent Control, Ministry of Education, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, 430074 Wuhan, China

\textsuperscript{2}Research Institute of Huazhong University of Science and Technology in Shenzhen, 518057 Shenzhen, China

* Corresponding author

Keywords: High confidence, Target tracking, Offline learning, Online updating.

Abstract. Target tracking is one of the most challenging tasks in computer vision. In this paper, the high confidence tracking (HCT) algorithm is proposed by combining the offline historical learning network with online correlation filter updating model. First, the weighted historical targets are introduced into the offline learning network, which solves the problem of target loss caused by inaccurate tracking of the previous frame. Second, the target’s confidence detection mechanism is proposed, and added to the correlation filter tracking algorithm, so that the model drift is avoided. Finally, we form a new high confidence tracking algorithm with offline learning. Compared with the state-of-the-art tracking algorithm, our algorithm performs outstandingly on benchmark OTB13 and OTB15, while ensuring real-time performance.

Introduction

Target tracking is a basic and challenging problem in computer vision. Trackers based on correlation filter (CF) have attracted wide attention in tracking field for two main reasons. Firstly, spatial correlation can be calculated in frequency domain, so tracking method based on CF is very suitable for fast tracking. Secondly, compared with sparse responses generated by most existing tracking-by-detection methods, dense responses in search areas can be generated using CF trackers. Therefore, CF trackers have achieved good results on popular tracking benchmark datasets.

The GOTURN algorithm [1] runs fast, but the target tracking accuracy is low. The kernelized correlation filter (KCF) [2] has an online update mechanism, but lacks an evaluation of the target’s confidence. In order to solve the above problems, a fast target tracking algorithm is proposed in this paper. Our contributions are summarized as follows: 1) We propose a more robust offline network architecture where the inputs include historical targets. 2) In order to prevent the model drift, we propose a new detection mechanism of tracking result confidence. 3) We propose a fusion mechanism, named as high confidence tracker (HCT), to fuse the offline historical learning with the online correlation filter updating.

Enhanced GOTURN

GOTURN’s network structure consists of a Siamese network and a regression network. The current frame and the previous frame are inputted to the Siamese network. The regression network is a fully connected layer. The position and the size of the current frame’s target can be regressed.

An enhanced GOTURN method is proposed to replace the target area in previous frame with a weighted image of the historical frame targets, and the background of the previous frame is unchanged. The network's input is a search area of the current frame, together with a joint target image which is merged from previous information. The network finds targets in the current frame by learning to compare these crops. The weighted image of the multi-frame target areas reduces the
influence of the inaccurate information of the previous frame target and improves the robustness of the network. The enhanced network framework is shown in Fig.1.

![Enhanced GOTURN Architecture](image)

**Figure 1. An enhanced GOTURN architecture.**

The network has two inputs, one is the target search area of the current frame, and the other is the weighted image of the historical target areas:

\[
target_{pad}^{t} = \rho \cdot target_{\text{pad}}^{t-1} + (1 - \rho) \cdot target_{pad}^{1}.
\]

(1)

\[
target_{pad}^{t} = target_{pad}^{t-1}.
\]

(2)

where \(target_{pad}^{t}\) is the target input in the \(t\)-th frame, \(target_{pad}^{1}\) is the region of interest in first frame, \(target_{\text{pad}}^{t-1}\) is tracking results of the previous frame image, \(target_{\text{pad}}^{t-1}\) is the target input in the \(t-1\)-th frame, \(\rho\) is weighting factor.

**High Confidence Updating Mechanism**

In LMCF [3], the degree of fluctuation of the response map is expressed by the average peak-to-correlation energy (APCE), and the calculation for the APCE is as follow:

\[
APCE = \frac{\left| F_{\text{max}} - F_{\text{min}} \right|}{\text{mean} \left( \sum_{w,h} (F_{\text{w,h}} - F_{\text{w,h}}^f) \right)},
\]

(3)

where \(F_{\text{max}}\), \(F_{\text{min}}\) and \(F_{w,h}\) respectively represent the maximum, minimum values and corresponding values of the \(w\) row and the \(h\) col pixel in the response map.

Whether the KCF model is updated is determined by Eq.(4).

\[
F_{\text{max}} \geq \beta_1 \cdot F_{\text{max}} & \& APCE \geq \beta_2 \cdot APCE.
\]

(4)

However, the high confidence updating mechanism using APCE has a disadvantage. Although the APCE of response map is small, the maximum point in response map is still the correct target center, and the model can be updated with the tracking results of the current frame. The reason for the large fluctuation reflected by APCE is that the influence of smaller peaks and valleys on the fluctuation degree is considered. The degree of oscillation indicates the oscillation of the output response relative to the minimum response. Therefore, we propose a new way to calculate the target credibility. Since the maximum point is selected as the new position of the target, the point with a small response value does not interfere with the selection of the target, and the influence coefficient of these points is suppressed to 0 when reliability is calculated. Meanwhile, the fluctuation degree is computed relative to the zero plane. The target credibility can be computed more accurately according to the method. It is called APME (average peak-to-median energy), and the Eq. (5) is defined as follow:
\[ APME = \frac{1}{\text{mean}} \sum_{x,y} F_{\text{max}}^2 \varepsilon(F_{\text{max}} - F_{\text{threshold}}) \] \quad (5)

where \( F_{\text{max}} \) is the peak value, \( F_{\text{min}} \) is the minimum value, and \( \varepsilon \) is the unit step function. The larger the APME is, the higher the credibility of the detected target is. Whether the KCF model should be updated can be determined according to the Eq. (6).

\[ F_{\text{max}} \geq \beta \varepsilon \cdot F_{\text{max}} - \varepsilon \cdot \text{APME} \geq \beta \varepsilon \cdot \text{APME}. \] \quad (6)

**High Confidence Tracker**

Our target tracking algorithm combines the Enhanced GOTURN algorithm with the KCF algorithm, and integrates the target credibility detection to effectively track the target, named as high confidence tracker (HCT). Our HCT algorithm is designed as shown in Fig. 2.

![Diagram of High Confidence Tracker](image)

Figure 2. Main steps of the proposed ensemble algorithm.

Two tracking results are obtained, which are KCF_Box and GOCF_Box. In this paper, the APME of response map is calculated as the basis for selecting the tracking result box. Besides, Numerous experiments have shown that the scale of the EnGOTURN_Box is unstable. Therefore, the scale of KCF_Box is retained as the final scale. The determination of Result_Box is shown below.

\[
\text{result \_Box\_center} = \begin{cases} 
\text{GOCF\_Box\_center} & \text{APME}_{1} < \text{APME}_{2} \\
\text{KCF\_Box\_center} & \text{APME}_{1} \geq \text{APME}_{2}
\end{cases}
\]

\[
\text{result \_Box\_scale} = \text{KCF\_Box\_scale}
\] \quad (7)

where \( \text{APME}_{1} \) and \( \text{APME}_{2} \) are the APME of the response map in KCF and GOCF respectively.

The inputs of enhanced GOTURN are updated in every frame, since if the background information is not updated for several consecutive frames, the result of the regression will be inaccurate.

Whether the tracking model of KCF is updated with the target result is judged according to the template updating strategy described in the Eq. (6) in Section 3. \( F_{\text{max}} \) and APME in Eq. (6) are selected according to the Eq. (8).

\[
(F_{\text{max}}, \text{APME}) = \begin{cases} 
(F_{\text{max}1}, \text{APME}_{1}), \text{APME}_{1} < \text{APME}_{2} \\
(F_{\text{max}2}, \text{APME}_{2}), \text{APME}_{1} \geq \text{APME}_{2}
\end{cases}
\] \quad (8)

where \( F_{\text{max}1} \) and \( F_{\text{max}2} \) are the maximum value of the response map in KCF and GOCF respectively.

**Experiments**

In order to verify the effectiveness of our proposed algorithm, a series of comparative experiments in benchmark OTB13 [4] and OTB15 [5] are conducted. OTB15 contains other 50 videos except 50 videos in OTB13. OPE (one-pass evaluation) [4] is used to evaluate our method.
Implementation Details

Videos and static pictures are used to train enhanced GOTURN network. The training data comes from ALOV300++ [6] and Images in the ImageNet Detection Challenge [6]. The weighting factor is set to 0.25 in our experiments. The feature used in the KCF algorithm is the HOG feature. The $\beta_1$ and $\beta_2$ in the KCF model update condition of Eq. (6) are 0.7 and 0.45 respectively. Our HCT algorithm is implemented with C++ and its Enhanced GOTURN runs on a single NVIDIA Geforce GTX TITAN X GPU with 16 GB memory and Intel Xeon® CPU E5 2630 v3 @ 2.40GHz with 256GB memory.

Baseline Comparison

In OTB13, we firstly test the improvement of Enhanced GOTURN over GOTURN, and then test the tracking results of APME and APCE applied to KCF, respectively labeled KCF with APME and KCF with APCE. In order to have a more intuitive evaluation of the tracking algorithm, DPR and OSR are used to evaluate the tracking precision and success respectively.

In Fig. 3, enhanced GOTURN has better accuracy and robustness in OPE compared with GOTURN. Due to the weighted historical frame targets as input, in a complex scenario, the network model can still regress an accurate target position. KCF with APME has the best tracking accuracy in OPE. KCF gets poor performance because of corrupted model. As the model is updated when the condition shown in Eq. (6) is met, the KCF with APME runs faster. The credibility of the target computed using APME is higher to satisfy the updating condition of the model, which causes that the tracking model can keep up with the DEF of the target. Therefore, the accuracy of KCF with APCE is less than KCF with APME.

![Precision plots of OPE](image1.png)  ![Success plots of OPE](image2.png)

Figure 3. Comparison among the proposed algorithm and its baselines.

Evaluation on HCT

Compared to the 26 state-of-the-art trackers from the CF trackers based on conventional features and CF trackers based on deep features, we evaluated the HCT on OTB15. For CFNet, we use the best results from its baseline-on-cfnet-conv3.

In Fig. 4, the proposed HCT performs better than most of CF trackers based on deep features. Our HCT ranks only third in terms of precision and success scores, it is not as good as ECO and CCOT trackers, but the speed of HCT is 9 times and 180 times faster than ECO and CCOT respectively. The running speeds of HCT and other state-of-the-art trackers are shown in Table 1 and Table 2. Poor real-time performance is a severe limitation of its application. Although siamese network is used in HCT, CFNet, and Siamese FC, and correlation filter is introduced in CFNet, the convolution of two feature maps in Siamese FC is similar to template matching in HCT, our HCT algorithm outperforms CFNet and Siamese FC in both distance precision rate and overlap success rate. This result shows the importance of online updating which can learn feature information of the target.
Table 1. Comparisons with state-of-the-art deep features based trackers on OTB15.

| trackers   | ECO | CCOT | Deep-SRDCF | Siamese-FC | CFNet | CF2 | ACFN | HDT | HCT |
|------------|-----|------|-------------|------------|-------|-----|------|-----|-----|
| speed/fps  | 6   | 0.3  | <1          | 58         | 75    | 11  | 15   | 10  | 54  |
| OSR(%)     | 70.4| 66.6 | 63.2        | 60.9       | 60.7  | 60.3| 60.1 | 60.0| 63.4|
| DPR(%)     | 84.2| 81.5 | 77.0        | 73.7       | 74.7  | 79.9| 78.1 | 79.5| 76.3|

Table 2. Comparisons with state-of-the-art conventional features based trackers on OTB15.

| trackers   | MCPF | BACF | ECO-HC | Staple_CA | LCT  | SRDCF | SAMF_AT | KCF | HCT |
|------------|------|------|--------|-----------|------|--------|---------|-----|-----|
| speed/fps  | 0.5  | 35   | 15     | 35        | 27   | 5      | 7       | 135 | 54  |
| OSR(%)     | 68.0 | 65.2 | 64.6   | 62.5      | 62.3 | 61.8   | 60.7    | 49.6| 63.4|
| DPR(%)     | 84.2 | 78.6 | 79.3   | 76.6      | 76.3 | 75.3   | 76.7    | 61.9| 76.3|

In Fig. 5, HCT is more accurate than most CF trackers based on conventional features. Although ECO-HC, MCPF, and SRDCFdecon are more accurate than HCT, their speed is much lower than that of HCT. The spatial regularization term is added to the SRDCFdecon as the penalty term for edge features, which solves the problem of edge effect. At the same time, alternating convex search ADS is used to optimize the tracking model coefficient, the possibility of the model being polluted is reduced, which is consistent with the aim of using APME to prevent the model drift in HCT, so the accuracy of HCT is very close to that of SRDCFdecon. BACF is similar to HCT on OSR, but its speed is not as fast as HCT. For the remaining CF trackers based on conventional features, HCT is superior to them in accuracy and speed. It can be seen from the figure 5, HCT achieves an OSR of 62% and a DPR of 74.8%, which is far superior to KCF as baseline.

Figure 5. The plots of OPE comparing HCT with the state-of-the-art conventional features based trackers.

In general, our HCT tracker performs better than the DeepSRDCF, CF2, HDT, and CFNet with convolutional features. Besides, HCT and DeepSRDCF get similar performance. Due to deep feature from Enhanced GOTURN and target credibility detection mechanism, success plots are increased by 13.8% and 19.5% compared with traditional KCF and GOTURN respectively. Although the overall performance of HCT is slightly lower than the ECO (6 fps) and MCPF (0.5fps), we can achieve competitive performance with faster speed.

Conclusion

In this paper, we present the HCT algorithm which combines conventional features and deep features with a target credibility detection mechanism. In order to enhance the robustness of the network and
improve the accuracy of the deep tracking algorithm, we propose enhanced GOTURN convolution network, so that the network is not affected by the inaccurate target. In order to prevent model drift, we propose a target credibility detection method for all CF trackers. At the same time, we design an ensemble algorithm, which effectively combines the enhanced GOTURN with the CF tracker having target credibility detection to form a new algorithm HCT. Finally, the baseline experiments are made on the OTB13, and the HCT algorithm is compared with the existing top trackers on the OTB15. Experiments show that our HCT algorithm not only has high speed, but also has high tracking accuracy, which is beneficial to many real-time applications.

Acknowledgement

This research was supported by National Natural Science Foundation of China under Grant No. 61105006; Shenzhen Strategic Emerging Industry Development Special Fund under Grant No. JCYJ20170307172130906; Aerospace Science and Technology Foundation under Grant No. 2018-HT-HZ; Open Fund of Key Laboratory of Image Processing and Intelligent Control (Huazhong University of Science and Technology), Ministry of Education under Grant No. IPIC2019-01; and the China Scholarship Council under Grant No. 201906165066.

References

[1] D. Held, S. Thrun, S. Savarese, Learning to track at 100 FPS with deep regression networks, ECCV. (2016) 749-765.
[2] J.F. Henriques, R. Caseiro, P. Martins, High-speed tracking with kernelized correlation filters, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2014) 583-596.
[3] M. Wang, Y. Liu and Z.Y. Huang, Large margin object tracking with circulant feature maps, CVPR. (2017) 4800-4808.
[4] Y. Wu, J. Lim, M.H. Yang, Online object tracking: a benchmark, CVPR. (2013) 2411-2418.
[5] Y. Wu, J. Lim, M.H. Yang, Object tracking benchmark, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2015) 1834-1848.
[6] A.W. Smeulders, et al., Visual tracking: an experimental survey, IEEE Trans. Pattern Anal. Mach. Intell. 36(2013) 1442-68.