An improved TLD algorithm with selective detection

WANG Jiao-yao¹,², HOU Zhi-qiang¹,², YU Wang-sheng¹, LIAO Xiu-feng¹ and CHEN Chuan-hua¹

¹ Institute of Information and Navigation, Air Force Engineering University, Xi’an 710077, China
² Institute of Computer, Xi’an University of Posts and Telecommunications, Xi’an 710121, China

Abstract. This paper proposes an improved TLD algorithm with selective detection. In the tracking module, KCF is used as the short-term tracker, and the idea of backward tracking is introduced to judge the accuracy of the result, when the result is inaccurate, the proposed algorithm starts up the detection module; in the detection module, H component of the color image is used as the input, and the mean filter serves as the first layer of the cascaded classifier, the color histogram similarity measure method is used to judge the accuracy of the detection results; finally a fusion strategy is formulated according to the output results of the tracking module and the detection module. The experiments carried on the OTB-2013 data platform show that the tracking accuracy and success rate of the proposed algorithm are respectively 0.801 and 0.591, which are 20.4% and 16.5% higher than the TLD algorithm, besides the proposed algorithm have good adaptability in scenarios such as illumination change, occlusion, and scale change, which shows superior tracking robustness.

1. Introduction

Nowadays visual tracking has become an important research contents of the computer vision [1-2], scholars at home and abroad have proposed a lot of tracking algorithms, which can be generally classified as generative tracking and discriminant tracking. Generative tracking algorithms represents the target through a model, and it searches the target by finding the area most similar to the model, the typical algorithms are Meanshift[3], IVT [4] and ASLA [5];Discriminant tracking algorithms regards tracking as a two-classification problem, whose purpose is to distinguish the target from the background, typical algorithms include SVT [6], CT [7], and MIL [8]. Most algorithms can achieve good tracking results in simple environments, but when affected by factors such as illumination variation, occlusion, and motion blur, these algorithms will suffer tracking drift. And this problem is particularly prominent in the long-term tracking.

In 2012 Kalal et al. proposed the TLD[10] algorithm, which combined tracking algorithms with detection algorithms, and continuously updates the target model through online learning, it provides a good tracking framework for long-term visual target tracking.

To achieve better robustness, this paper proposes an improved TLD algorithm with selective detection. In the proposed algorithm, the tracking module uses the Kernelized Correlation Filter (KCF) algorithm [11] and a backward tracking method is used to judge whether the target is missing, if the target is lost, the detector is started to relocate the target; in the detection module the image is converted to HSI space, and the H component of the image is extracted as the input, the mean
classifier is employed instead of the variance classifier to better filter the background window. The proposed algorithm was tested on 50 sets of video on the OTB2013 data platform, and the comparison results with the other seven current mainstream algorithms from both the qualitative and quantitative aspects, verified the superiority of the proposed algorithm.

2. The proposed algorithm

In order to improve the tracking robustness and real-time performance of TLD algorithm, this paper improves it from the tracking module and the detection module, and makes corresponding adjustments in the fusion strategy.

2.1 Tracking module

The specific flow of the tracking module in the proposed algorithm is shown in Figure 1. When the \( k \)th frame arrives, the KCF algorithm uses the \( (k-1) \)th frame tracking result \( p_{k-1} \) and filter model \((\partial_{k-1}, x_{k-1})\) to calculate the position of the target in the current frame, after that it calculates the filter model \((\partial_{curr}, x_{curr})\) of the current frame according to its result \( p_{k} \). In order to determine whether the tracking result is accurate, a backward tracking mechanism is added in the tracking module, that is backward tracking the target in the \( k-1 \) frame by using \( p_{k} \) and the learned filter model \((\partial_{curr}, x_{curr})\), after obtaining the estimate position \( p_{k-1}^{est} \) of the target in the \( (k-1) \) th frame, the error between \( p_{k-1} \) and \( p_{k-1}^{est} \) is calculated. The accuracy of the tracking module results is determined by comparing the error and the threshold \(_{thr}T_1\).

We found that if the threshold is too small, the detection module will be too sensitive; if the threshold is too large, the detection module will be too dull. Therefore, the proposed algorithm set the threshold adjustably. First, the target is divided into three categories. The classification rules are as follows: first finding the short side and long side of the target, respectively marked as \( \min = \min \{w, h\} \) and \( \max = \max \{w, h\} \), and then calculating the ratio \( r = \min/\max \), if \( 0 < r <= 1/3 \), the target is classified as type I, if \( 1/3 < r <= 2/3 \), the target is classified as type II, if \( 2/3 < r <= 1 \), the target is classified as type III. On the OTB2013 data platform, the number of Type I, Type II, and Type III targets are respectively 8, 12, and 30, and the threshold is calculated as shown in Equation (1). In order to determine the value of \( \delta \), this paper carried out experiments by setting \( \delta \) as \( \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7\} \), and the average center location error was used as the evaluation criterion. The experimental results are shown in Table 1, from the table we can find that for the target of Type I, Type II and Type III, when the value of \( \delta \) correspondingly set as 0.3, 0.4 and 0.6, the tracking performance is the best.

\[
_T\_thr = \frac{1}{2}(\delta \max + (1-\delta) \min) \tag{1}
\]

| Type | \(\delta\) | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|------|--------|-----|-----|-----|-----|-----|
| I(8) |       | 11.9| 18.6| 33.3| 36.4| 36.6|
| II(12)|      | 21.6| 20.3| 28.9| 41.6| 44.8|
| III(30)|     | 18.9| 20.1| 22.8| 16.6| 30.7|

Table 1. Tracking results comparison under different values of \( \delta \)
2.2 Detection module

The proposed algorithm uses a three-level cascading classifier to detect the target, and converts the image to HSI space, it uses the H component as the input of the cascaded classifier.

The three-level cascade classifier of the proposed algorithm includes mean classifier, ensemble classifier and nearest neighbor classifier, the first classifier of the cascade classifier plays the role of filtering a large number of background windows. However in the TLD algorithm the filtering effect of the variance classifier is not obvious, when the target is simple in shape. So the mean classifier is used in this paper as the first classifier. The detection steps are as follows: firstly, when the algorithm is initialized, calculating the mean $\mu_0$ and standard deviation $\sigma_0$ of the target image, considering that the H component of the image is not sensitive to changes in the light, so it can be assumed that the H component of all target windows conforms to a normal distribution, whose mean is $\mu_0$ and variance is $\sigma_0$. If the mean value of the window is within the range of $[\mu_0 - 3\sigma_0, \mu_0 + 3\sigma_0]$, it is considered as the target window. If not, it is considered as the background window.

3. Experimental results

The proposed algorithm was tested on the platform of Matlab 2014b and Visual Studio 2013. The hardware environment was Intel(R) Core i7 2.8GHz. The experimental test data is 50 sets of video...
sequences provided by OTB2013, and comparison was made with 7 mainstream tracking algorithms, including Staple$^{[12]}$, DSST$^{[13]}$, KCF, DLT$^{[14]}$, TGPR$^{[15]}$, Struck$^{[16]}$, and TLD.

3.1. Qualitative analysis

Qualitative analysis was made from the following six aspects:

**Deformation**: In the video sequences of Basketball and Bolt, the target undergoes local non-rigid deformation during rapid movement. It can be seen that only the proposed algorithm can keep tracking the target. **Occlusion**: In the video sequences of Walking2 and FaceOcc2, the target is partially occluded. Only the proposed algorithm and the DSST algorithm can always track the target accurately. In the video sequence of Jogging-1, the target is completely occluded, when it reappears, only the proposed algorithm and the Staple algorithm can capture the target immediately.

**Illumination variation**: In the video sequences of Singer1 and CarDark, the background light changes drastically. It can be seen from Fig. 3(e) and Fig. 3(f) that only the proposed algorithm can accurately track the target.

**Scale change**: In the video sequence of Doll, as shown in Fig. 3(g), the scale of the target first increases and then decreases. It can be seen that only the proposed algorithm can adapt to the scale change of the target well, in the video sequence of Walking2, the target has both scale changes and occlusion in the movement process. Most of the algorithms can keep track of the target but only the tracking results of the proposed algorithm is the most accurate.

**Rotation**: In the video sequence of FaceOcc2, the target rotates in the image plane, the proposed algorithm is able to maintain stable tracking. In the video sequence of Skiing, the target not only has rotation but also moves faster. It can be seen from Figure 3(j) that only the proposed algorithm can keep tracking of the target.

**Low resolution**: In the video sequences of Freeman4 and Skiing, most of the algorithms lose the targets due to the small size of the target, besides there are other interferences such as occlusion and fast motion. Only this algorithm can continuously track the target.
3.2 Quantitative analysis

In order to evaluate the tracking performance of the algorithm more comprehensively, this paper quantitatively analyzes the algorithm from two aspects: single video and 50 sets of video.

**Quantitative analysis under single test video**

For the 10 video sequences shown in Fig. 3, the tracking performance of the algorithm is analyzed by using two indicators: average center position error and average overlap ratio. The center position error refers to the euclidean distance between the tracking position and the true position, and the overlap ratio refers to the ratio of the intersection area and the union area of the tracking result and the groundtruth.

Table 2 and Table 3 respectively show the comparison of the average center position error and the average overlap ratio of all algorithms under these 10 groups of test videos. The optimal result of each test video in the table is marked in red, and the suboptimal results is marked in blue,"-" means tracking failed. It can be seen from the table that the algorithm of this paper generally has a lower average center position error and a higher average coverage rate, which exhibits good tracking performance.

![Figure 3. The comparisons of the tracking algorithms in different attributes](image)

| Staple | DSST | KCF | DLT | TGPR | Struck | TLD | Ours |
|--------|------|-----|-----|------|--------|-----|------|
| Basketball | 9.7  | 111.6 | 8.1 | 12.0 | 9.4  | 118.3 | — | 8.9 |
| Bolt | 17.3 | 6.0 | 6.9 | — | 424.1 | 398.2 | — | 5.8 |
| Singer1 | 9.2 | 3.8 | 12.6 | 3.4 | 119.9 | 14.5 | 7.9 | 6.1 |
| CarDark | 4.9 | 1.0 | 5.8 | 19.1 | 2.1 | 0.9 | 27.5 | 2.98 |
| Walking2 | 9.9 | 3.2 | 29.5 | 2.0 | 5.7 | 11.1 | — | 2.9 |
| Doll | 4.9 | 2.7 | 8.2 | 5.8 | 5.9 | 8.8 | — | 2.4 |
| FaceOcc2 | 7.1 | 6.7 | 7.7 | 11.4 | 7.5 | 5.9 | 12.3 | 5.9 |
| Jogging-1 | 3.9 | 111.8 | 87.6 | 112.8 | 136.8 | 62.1 | 6.7 | 2.6 |
| Freeman4 | 9.9 | 5.5 | 26.9 | 45.1 | 48.0 | 49.2 | — | 10.9 |
| Skiing | 8.1 | 219.7 | 259.6 | 243.2 | 282.9 | 251.7 | — | 5.9 |

| Staple | DSST | KCF | DLT | TGPR | Struck | TLD | Ours |
|--------|------|-----|-----|------|--------|-----|------|
| Basketball | 73.9 | 28.4 | 67.1 | 50.7 | 70.1 | 19.7 | — | 69.8 |
| Bolt | 69.0 | 75.2 | 67.1 | — | 1.7 | 1.5 | — | 80.8 |
| Singer1 | 35.1 | 40.9 | 35.6 | 84.1 | 41.2 | 36.2 | 72.6 | 77.6 |
| CarDark | 67.6 | 90.3 | 63.3 | 60.6 | 86.4 | 88.7 | 43.5 | 67.2 |
| Walking2 | 67.5 | 56.8 | 64.0 | 82.5 | 70.7 | 68.5 | — | 77.4 |
| Doll | 56.8 | 56.5 | 44.2 | 85.7 | 61.1 | 57.4 | — | 74.3 |
| FaceOcc2 | 75.6 | 80.6 | 74.7 | 60.4 | 72.7 | 78.7 | 57.0 | 80.2 |
| Jogging-1 | 73.9 | 18.8 | 19.8 | 18.7 | 20.5 | 16.6 | 80.4 | 84.8 |
| Freeman4 | 50.0 | 51.4 | 19.3 | 13.4 | 28.1 | 18.6 | — | 48.1 |
| Skiing | 42.9 | 6.3 | 6.2 | 8.5 | 9.1 | 4.6 | — | 56.3 |
Quantitative analysis under 50 sets of test video

In this section, the accuracy curve and success rate curve are introduced to evaluate the tracking performance on the 50 sets of test videos of the algorithm, the precision curve describes the ratio between the number of video frames whose center position error does not exceed a given threshold and the total number of frames. The success rate curve describes the ratio between the number of video frames whose overlap rate is greater than the given threshold and the total number of frames. The ground truth of the target in the above calculations is provided by the data platform. When evaluating the accuracy and success rate of this algorithm, the corresponding thresholds are set to 20 pixel and 0.5 respectively.

As shown in Figure 4, we can see that the algorithm achieved the best tracking accuracy of 0.801, and the suboptimal success rate of 0.591. Compared with the TLD algorithm, the proposed algorithm improves the accuracy and success rate by 20.4% and 16.5% respectively. In terms of tracking speed, owing to the selective detection, the tracking speed is much higher than the TLD algorithm. The experimental results show that the average tracking speed of the proposed algorithm reaches 34.9fps, which shows its real-time processing capability.

4. Conclusion

In this paper an improved TLD algorithm with selective detection is proposed. In the improved algorithm, the KCF algorithm is used as a short-term tracker of the tracking module, and the backward tracking method is introduced to determine the accuracy of the results of tracking module, only when the tracking result is inaccurate, the algorithm employs the detection module to refind the target. In the detection module, H component of the image is used as the input, and the variance filter is replaced with the mean filter to improve the detection efficiency of the detection module. The algorithm of this paper was tested on 50 sets of video sequences, and both the qualitative and quantitative analysis of the experimental results show that the proposed algorithm has good tracking performance and superior real-time performance.

During the experiment, it is found that under the situations of clutter background or similar object interference the performance of the proposed algorithm is not very prominent. And in the future, to further improve the robustness of long-term tracking, more research will be done to solve this problem.

References

[1] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, et al. Visual tracking: an experimental survey. IEEE T. P. A. M. I., 36, 24(2013)
[2] N. Y. Wang, J. P. Shi, D. T. Yeung, Understanding and diagnosing visual tracking systems(IEEE ICCV, Santiago, 2015)
[3] D. Comaniciu, V. Ramesh, P. Meer, Kernel-Based Object Tracking, P. A. & M. I., 25, 11(2003)
[4] D. A. Ross, J. Lim, R. S. Lin, Incremental Learning for Robust Visual Tracking. I. J. C. V., 77, 16(2008)
[5] X. Jia, H. Lu, M. H. Yang, Visual tracking via adaptive structural local sparse appearance model (IEEE CCVPR, 2012)
[6] S. Avidan, Support vector tracking. IEEE T. P. A. M. I., 26, 8(2004)
[7] K. Zhang, L. Zhang, M. H. Yang, Real-Time Compressive Tracking, (ECCV, Springer-Verlag, 2012)
[8] B. Babenko, M. H. Yang, S. Belongie, Robust Object Tracking with Online Multiple Instance Learning, IEEE T. P. A. & M. I., 33, 13(2011)
[9] Y. Wu, J. Lim, M. H. Yang. Online object tracking: A Benchmark, (IEEE CS CCVPR, USA, 2013)
[10] Z. Kalal, K. Mlikolajczy, J. Matas, Tracking- Learning-Detection, IEEE T. P. A. M. I., 34, 13(2012)
[11] J. F. Henriques, R. Caseiro, P. Martins, High- speed tracking with kernelized correlation filters, IEEE T. P. A. M. I., 37, 13(2015)
[12] L. Bertinetto, J. Valmadre, S. Golodetz, Staple: complementary learners for real-time tracking, (IEEE CVPR, 2016)
[13] B. H. Zhang, H. C. Lu, Z. Y. Xiao, Visual tracking via discriminative sparse similarity map, IEEE T. I. P., 23, 9(2014)
[14] N. Y. Wang, D. Y. Yeung, Learning a deep compact image representation for visual tracking, A. N. I. P. S., 8(2013)
[15] J. Gao, H. B. Ling, W. M. Hu, Transfer learning based visual tracking with gaussian processes regression, (Computer Vision – ECCV, Springer International Publishing, 2014)
[16] S. Hare, S. Golodetz, A. Saffari, Struck: structured output tracking with kernels, IEEE T. P. A. M. I., 23, 7(2015)