Proceedings of the 2017 International Conference on Machine Learning and Cybernetics, Ningbo, China, 9-12 July, 2017

ROBUST VISUAL TRACKING BASED ON L1 EXPANDED TEMPLATE

DANSONG CHENG[1], YONGQIANG ZHANG[1], FENG TIAN[2], DAMING SHI[3], XIANGFANG LIU[4]

[1]The School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, P.R.China
[2]The Faculty of Science & Technology, Bournemouth University, UK
[3]The School of Computer Science and Software Engineering, Shenzhen University
[4]The School of Electrical Engineering and Automation, Harbin Institute of Technology, P.R.China
E-MAIL: cdsinhit@hit.edu.cn

Abstract:

Most video tracking algorithms including L1 tracker often fail to track correctly under adverse conditions such as object occlusion, disappearance, etc. To address this issue, we propose an improved L1 tracker algorithm called Tracker-2, based on what we call the expanded template which includes the reference template and trail template. The reference template keeps the original features of the target and prevents errors from being introduced by false tracking results with the template update, which leads to the deviation of the target. The trail template records the trail tracking results to avoid massive use of trivial templates which may result in the false detection of occlusion. The experimental results on a number of standard data sets have proved that our Tracker-2 approach is able to deal with the occlusion problem effectively while maintaining the advantages of L1 tracker.

Keywords:

Visual Tracking; Sparse Representation; L1 Tracker; Reference Template; Trail Template

1. Introduction

Visual tracking has been one of the key research areas in computer vision community for the past decades. It is crucial for video analysis, surveillance and monitoring, human behavior analysis, human computer interaction and video indexing/retrieval etc.[1]. A number of approaches have been proposed for visual object tracking. Most of them can be classified into two categories: generative and discriminative. The generative methods [2, 3] aim at building the appearance model of target. The tracking is achieved by searching the location which is most similar to the learned appearance model. Different from the generative, the discriminative approaches solve a binary classification problem for tracking. Features are extracted from both the target and background regions, used for learning a classifier online, such as ensemble tracking [4]. There are also some other advanced theories used for tracking, including multiple instance learning [4, 5], sparse bayesian learning [6] , global model seeking [7] and Tracking-Learning-Detection (TLD) [8]. Recently, the sparse representation and compressed sensing techniques [9] have drawn a great attention in visual tracking [10, 11, 12]. In this case, the tracker represents each target candidate as a sparse linear combination of dictionary templates that can be dynamically updated to maintain an up-to-date target appearance model. This representation has been shown to be robust against partial occlusions, which leads to improved tracking performance.

The L1 tracker based approaches have addressed the occlusion, corruption and other challenging issues through a set of trivial templates. Specifically, to track the target in a new frame, each target candidate is sparsely represented in the space spanned by target templates and trivial templates. The sparsity is achieved by solving an L1-regularized least squares problem. Then the candidate with the smallest projection error is regarded as the target. Despite of being effective, these L1 trackers require high computational costs due to numerous calculations for L1 minimization. In addition, the inherent insensitivity of the L1 minimization to occlusion has not been fully utilized. Thereafter [11] proposed a modified efficient L1 tracker with minimum error bound and occlusion detection, called Bounded Particle Resampling (BPR)-L1 tracker. This algorithm is able to deal with occlusion with L1 minimization formulation using trivial templates, at the expense of high computational cost,though.

But if the image is polluted by complex noise or occlusion, L1 tracker method has some drawbacks in tracking accuracy and erroneous judgement of occlusion. To address this challenge, in this paper, we propose an improved L1 tracker ap-

978-1-5386-0408-3/17/$31.00 ©2017 IEEE

proach, which we call Tracker-2, by introducing the reference template and the trail template into the L1 tracker. The template of target extracted from the first frame is taken as the reference template, which is not updated during tracking. This can avoid errors from false tracking results and improve the recapture of the target after occlusion. According to the order of similarity, a template from recent frames is picked up as the trail template, which has the highest similarity with the target template.

2 L1 tracker algorithm

The L1 tracker proposed by Mei et al. [10] is a sparse-represented image tracking algorithm based on particle filtering. The sparse representation is used to calculate the likelihood p of a candidate object x_k. Given the target template set $T_k = \{t_k^1, t_k^2, ..., t_k^i, ..., t_k^n\}$, each template t_k^i is a 1D vector formed by stacking template image columns. The sets of candidate objects and the corresponding images are denoted as $X_k = \{x_k^1, x_k^2, ..., x_k^N\}$ and $Y_k = \{y_k^1, y_k^2, ..., y_k^N\}$, respectively [**?**]. A trivial template $I = \{i^1, i^2, ..., i^N\}$ is used to capture the occlusion. The tracking result y_k^i approximately lies in the linear span of T

$$y_k^i = T_k a_T^i + I_k a_I^i \ \ \forall y_k^i \in Y_k \tag{1}$$

where $a_T^i = (a_1, a_2, ..., a_n)^T$ is called a target coefficient vector, and a trivial template, I_k^i is a vector with only one nonzero entry (i.e. for the jth trivial template, only the jth dimension is 1 and all the other dimensions are 0). The coefficient $a_k^i = [a_T^i, a_I^i]$ is sparse, meaning that most of its dimensions are 0 or close to 0 except for a few dimensions. The definition of the Eq. (1), to a certain extent, illustrates the core idea of the sparse representation, i.e. finding the linear representation of the tracking target by feature templates and trivial templates. The error caused by occlusion and noise typically corrupts a fraction of the image pixels. Just for a candidate approximating the target, there are only a limited number of nonzero coefficients in a_k^i that account for the noise and partial occlusion. The deviation of the candidate from the target will reduce the sparsity of the coefficient vector, and increase the complexity of operation. Mei et al [11] exploited the compressibility in the transform domain by solving the problem as an L1-regularized least squares problem, which is known to yield sparse solutions typically.

$$\min \frac{1}{2} \parallel y_k^i - Aa \parallel_2^2 + \lambda \parallel a \parallel_1 \ \ a \geq 0 \tag{2}$$

3 Track-2 Algorithm

In the L1 tracker algorithm [10, 12], the occlusion is detected before template updates. In occlusion detection, the coefficient of the trivial templates, $a = [c_T, c_I]$ is converted into a matrix with the same size as the template, and then thresholded into a binary matrix with a predetermined threshold value. The largest connected region is used to estimate the occlusion. c_T is constrained to be larger than 0, but there is no constraint for c_I. That means, in the coefficient matrix, only those elements greater than the threshold are set to 1. When the gray value of an occlusion object is less than the tracking object, the occlusion will be ignored and not processed, which leads to the wrong tracking result. In order to address this issue, we propose an improved L1 tracker algorithm. At first, with a predetermined threshold value we threshold the absolute value of the coefficient matrix into a binary matrix. The elements with the equivalent value are regarded as in homogeneous regions. Then we calculate the largest connected area and compare it with the preset value given as a percentage of the total area (we set 30% in this paper).

To deal with the above-mentioned issues, we process the positive and negative templates separately, and introduce expanded templates (including the reference template and the trail template) into in addition to feature templates and trivial templates. As described earlier, L1 tracker applies a dynamic template-updating strategy interposing between frequently updating strategy and never updating strategy, while our reference template and trail template correspond to these two extreme strategies. We denote this approach as Tracker-2, which utilizes the particle filter and the flexibility of the sparse representation model to improve response capabilities of the tracking algorithm to track changes in appearance of the targets. Tracker-2 is also able to capture lost target and to reduce misdiagnosis of occlusion. Experimental results demonstrate that Tracker-2 can not only achieve better performance in occlusion detection than L1 tracker, but also prevent the template from being updated late due to overly strict occlusion detection.

3.1 Reference template

During the object tracking process, the target maybe occasionally disappear temporarily from the image sequence. Neglecting this case may weaken the ability of the algorithm to cope with object disappearance and to effectively recognize the target when it reappears [14]. For those tracking approaches [14] on responding dynamically to object changes, the disappearance and reappearance of the target causes more severe

problems. When the target disappears in the current frame, the approaches detect a big difference between the frame and the target template. This difference may be treated as a signal that the objective has changed. The target templates fail to respond appropriately to the changed objects but are updated according to the current wrong tracking results with some fallacious features being added into them. The error may eventually lead to offset of the tracking target.

3.2 Trail template

L1 tracker is prone to error on occlusion detection in object tracking as the template cannot describe adequately the current tracking process. The templates are not updated as needed, and the usage of trivial templates is reduced, which seriously affect the effective tracking.

To address this issue, we introduce the trail template, which is the tracking result with maximum similarity to the feature template among the latest results (the result of the last 5 frames, as in [15]). The trail template is updated with the latest and most reliable result in current process. Therefore, the trail template is the tracking result with two important characteristics reliable and recently-obtained. Firstly, it must be the target for its qualification as a template. Secondly, it is the result close to the current frame, so its difference from the current target is less than that between the current target and the feature template.

Here we consider a typical scenario previously mentioned for the false identification of occlusion. When small changes begin to appear, L1 tracker considers the template sufficient to describe the results of the current frame, thus does not update the template (If the template is updated, errors will be added into the template, resulting in a deviation in the tracking results). As the target gradually changes, a large number of trivial templates will be used when L1 tracker identifies the template incapable to describe the current frame appropriately. This may result in a large connected region with 1 in the binarized coefficient matrix, which will be identified as occlusion. Tracker-2 uses the trail template (latest matrix) to record the changes during the tracking process. As the templates characterize the gradual variation of the target, the number of trivial templates can be reduced significantly when the template is updated. So the risk of false detection of occlusion is reduced.

The overall process of the Tracker-2 algorithm is shown in Table I.

4 Experiment Results and Performance Analysis

In our experiments, Tracker-2 is implemented in Windows 7 using Matlab. The threshold of Sim [15] is preset to $\tau = 0.3$ (because the template is unable to represent the current tracking results when the value τ is more than 0.3.). The numbers of feature templates and particles are 10 and 600, respectively. $\lambda = 0.01$, $\mu_c = 10$ (Weight coefficient in Table I). The L1 tracker algorithm in the comparative experiment uses the same settings.

TABLE 2. APG method of L1 tracker[**?**]

Initialize $\alpha_0 = \alpha_1 = 0 \in \Re^N$,$t_0 = t_{-1} = 1$, $k = 0$;
While Non convergence do
$\beta_{k+1} := \alpha + \frac{t_{k-1}-1}{t_k}(\alpha_k - \alpha_{k-1})$;
$g_{k+1|_T} = \beta_{k+1|T} - (A^{'T}(A'\beta_{k+1} - y))\,|_T\,/(L - \lambda 1_T)$;
$g_{k+1|_I} = \beta_{k+1|I} - (A^{'T}(A'\beta_{k+1} - y))\,|_I\,/(\mu\lambda_{k+1|I}/L)$;
$\alpha_{k+1|_T} = \max(0, g_{k+1|_T})$;
$\alpha_{k+1|_I} = \sigma_{\lambda|L}(g_{k+1}\,|_I)$;
$t_{k+1} := \frac{1+(1+4t_k^2)^{1/2}}{2}$, $k = k + 1$
End

4.1 Comparative analysis of experimental results

In our quantitative experiments, 15 standard videos are chosen. Here we only randomly show the results of 4 videos. Tracker-2 is evaluated and compared with the L1 tracker.

To evaluate the tracking performance, Ref. [16] select the area-overlap-rate to determine whether each frame of the tracking results are acceptable. The overlap rate is the standard of the performance evaluation of object detection algorithm in PASCAL test [17], which is described as the overlap rate between the tracking target area and the actual target area. If the tracking target area in a frame is ROI_G, the actual target area is ROI_T, the overlap ratio of the frame is calculated by

$$SCORE = \frac{area(ROI_G \cap ROI_T)}{area(ROI_G \cup ROI_T)} \tag{3}$$

where N_{total} is the number of total frames, and $N_{score \geq 0.5}$ denotes the number of correctly tracked frames.

In addition to the above method, some researches [15] also apply the central discriminant method to determine whether the current tracking results meet the requirements, in which the distance between the centres of two regions is used as the dominate parameter. But this method ignores the size of the selected area, and is not accurate enough. So in our study we use the overlap rate approach, for it gives more accurate and comprehensive results. The comparison between results of Tracker-2 and L1 tracker is shown in Table III. Regarding the accuracy rate, our Tracker-2 has obtained similar performance to L1

TABLE 1. Tracker-2 Algorithm

Input: template set $T_k = \{T_1, T_2, \cdots, T_n, T_{reference}, T_{trail}\}$; Image F_k ; Particle set $S_{k-1} = \{x_{k-1}^i\}_1^N$; Occlusion timer μ_c
Output: template set T_k ; tracking result x_k^* ; Particle set $S_k = \{x_{k-1}^i\}_1^N$; Occlusion timer μ_c

Update μ according to μ_c , if $\mu_c > 0$, $\mu_c = \mu_c - 1$
Determine the candidates of this frame, x_t^i from the resulting particles of last frame x_{k-1}^i
for i = 1, $i < n$ do
create the corresponding image y_k^i from x_t^i
calculate the current observation probability q_i
end
sort the candidates according to q, set $i = 0$, $\tau = 0$;
while $i < N$ and $q_i \geq \tau$ do
perform the APG method for Tracker L1 (as shown in Table II), seeking the answer of minimization equation
calculate p_i.
$\tau = \tau + \frac{1}{2N} p_i$, $i = i + 1$;
end
$\forall j \geq i$, set $p_j = 0$;
Find the optimal x_k^* and its serial number t
if $\mu_c = 0$, $Sim(y_k^*, t_j) > \tau$ then
Calculate positive or negative coefficient matrix, configure μ_c ;
If no occlusion is detected, update the template;
end
Try to update the Trail template
Process the candidate x_k^* in accordance with the current p_i,generating particles x_k^*

tracker in processing the three datasets: Car, Singer, and Walking. This indicates that Tracker-2 still possesses the advantages of L1 tracker. For the Face dataset, Tracker-2 achieves a significant increase (almost double) in overall accuracy rate compared to L1 tracker. In summary, through the analysis of the results of four test sets, it can be seen that proposed method improves the tracking performance while in the meantime maintains the advantages of L1 tracker.

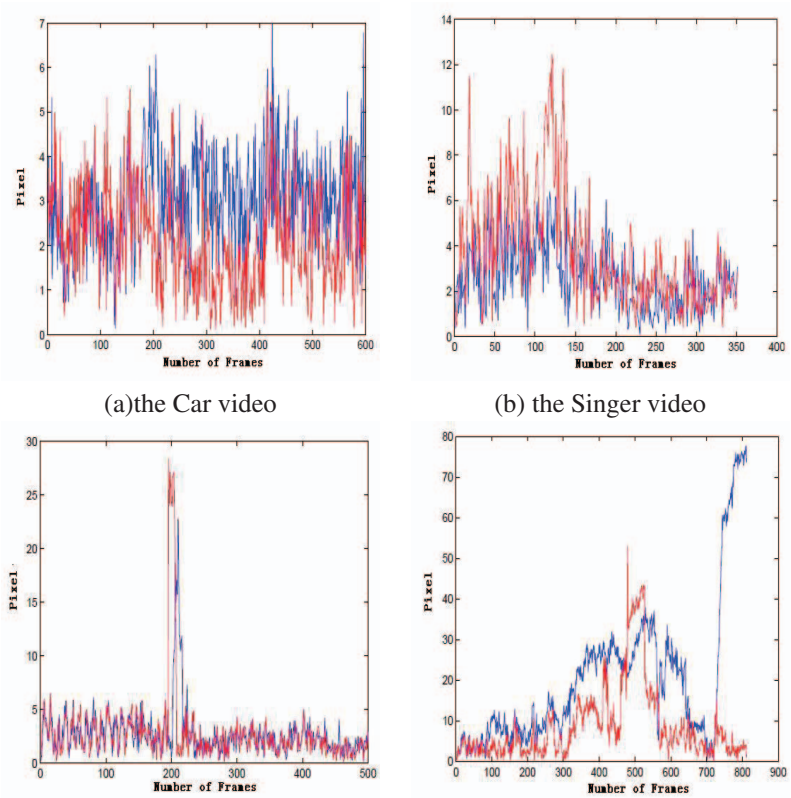
TABLE 3. Comparisons of accuracy rate for two methods

Index	L1 tracker	Tracker-2
Car	0.9883	0.9967
Singer	0.9952	0.9972
Walking	0.9800	0.9842
Face	0.2931	0.5727

To further evaluate the performance of Tracker-2, we compare the tracking results in the corresponding frames of two methods with the central moment of the standard target position, as shown in Fig. 1. The blue and red are the test results of L1 tracker and Tracker-2, respectively. The vertical axis represents the central moment of the tracking results and the actual target position (the unit is the pixel distance), and the horizontal coordinates represent the number of current frame.

4.2 Comparative analysis

In this part, we give intuitive comparisons of the image frames and the tracking results to illustrate Tracker-2's effectiveness. In Figure 2, the left is the initial calibration frames, followed by the recognition of different times. The red box encloses the tracking results of L1 tracker. In the Face dataset, the tracking accuracy is greatly affected by the appearance of a large number of occlusions which make the target disappear time to time. In Fig. 2(b), the results of L1 tracker are obviously shifted starting from the deflection of target's head. In Fig. 2(a), Tracker-2 can complete the tracking task, and there is no obvious deviation. Especially in the last stage of tracking, although the target box of Tracker-2 becomes relatively small, its location is still very accurate. This shows that , Tracker-2 copes with the occlusion more effectively than L1 tracker.



(a)the Car video (b) the Singer video

(c)the Walking video (d) theFace video

FIGURE 1. Test results of the video. Blue:L1 , Red:Tracker-2

5. Conclusions

In this paper an improved L1 tracker algorithm, Tracker-2, is proposed. We have proposed in this paper a new approch, called Tracker-2, using the expanded template (including the trail template and reference template) to deal with the problem of occlusion and misdiagnosis in image target tracking. We have evaluated and tested Tracker-2 on some classical video datasets and compared it with recent tracker algorithms. The experimental results have demonstrated that the expanded template strategy has effectively improved its ability to cope with the problem of occlusion, while retaining the advantages of conventional L1 tracker. In future work, we will try introducing the low rank representation into the algorithm to improve the accuracy of Visual Tracking, and apply the Tracker-2 algorithm to target track in complex environment

Acknowledgements

This research was supported by the National Natural Science Foundation of China (Grant No. 61402133)

References

[1] A. Yilmaz, O. Javed, and M. Shah, Object tracking: A survey, 2006.

[2] B. J. Yves, Bouguet. pyramidal implementation of the lucas kanade feature tracker, in Intel, 2010.

[3] I. Leichter, Mean shift trackers with cross-bin metrics, Pattern Analysis Machine Intelligence IEEE Transactions on, vol. 34, no. 4, pp. 695C706, 2012.

[4] B. Babenko, M. H. Yang, and S. Belongie, Visual tracking with online multiple instance learning, in Conference

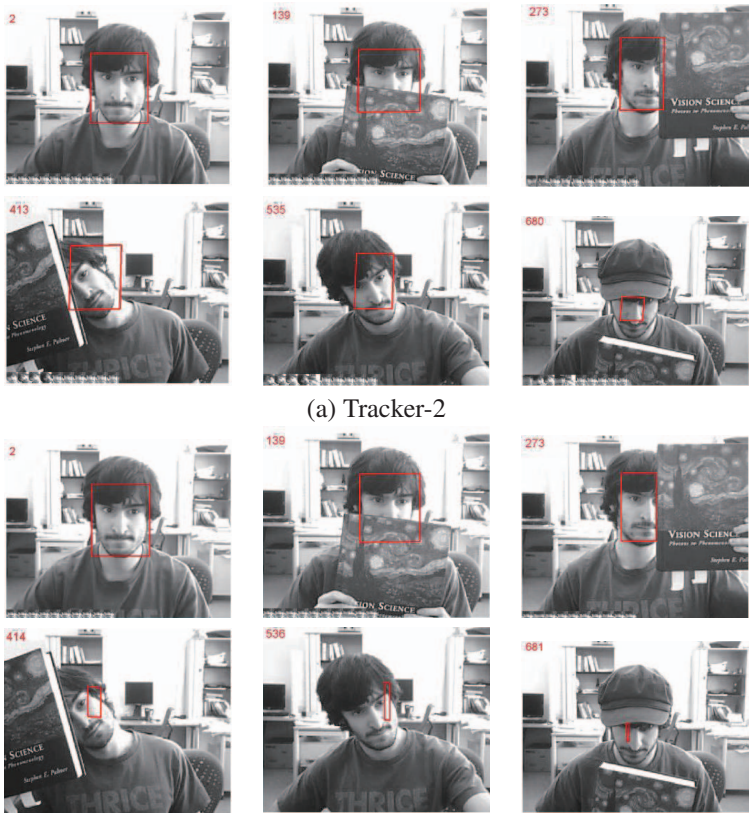


FIGURE 2. Comparison of tracking in the Face video.

.

on Computer Vision and Pattern Recognition, 2009, pp. 983C990.

[5] K. Zhang and H. Song, Real-time visual tracking via online weighted multiple instance learning, Pattern Recognition, vol. 46, no. 1, pp. 397C411, 2013.

[6] C. Sminchisescu, A. Kanaujia, and D. N. Metaxas, Bme : Discriminative density propagation for visual tracking, Pattern Analysis Machine Intelligence IEEE Transactions on, vol. 29, no. 11, pp. 2030C2044, 2007.

[7] Z. Yin and R. T. Collins, Object tracking and detection after occlusion via numerical hybrid local and global mode-seeking, in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, 2008, pp. 1C8.

[8] Z. Kalal, K. Mikolajczyk, and J. Matas, Tracking-learning-detection, Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 34, no. 7, pp. 1409C1422, 2012.

[9] H. Li, C. Shen, and Q. Shi, Real-time visual tracking using compressive sensing, in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011, pp. 1305C1312.

[10] X. Mei and H. Ling, Robust visual tracking using l1 minimization, in Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009, pp. 1436C1443.

[11] S. Zhang, X. Lan, H. Yao, H. Zhou, D. Tao, and X. Li, A biologically inspired appearance model for robust visual tracking, 2016.

[12] C. Bao, Y. Wu, H. Ling, and H. Ji, Real time robust l1 tracker using accelerated proximal gradient approach, in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, pp. 1830C 1837.

[13] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, Robust visual tracking via structured multi-task sparse learning, International journal of computer vision, vol. 101, no. 2, pp. 367C383, 2013.

[14] H. Zhou, T. Liu, J. Z. Zheng, F. Lin, Y. Pang, and J. Wu, Tracking nonrigid objects in video sequences, International Journal of Information Acquisition, vol. 3, no. 02, pp. 131C137, 2006.

[15] W. Zhong, H. Lu, and M.-H. Yang, Robust object tracking via sparsitybased collaborative model, in Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, pp. 1838C1845.

[16] Z. Kalal, J. Matas, and K. Mikolajczyk, Pn learning: Bootstrapping binary classifiers by structural constraints, in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010, pp. 49C56.

[17] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, The pascal visual object classes (voc) challenge, International journal of computer vision, vol. 88, no. 2, pp. 303C338, 2010.