Object Relocation Visual Tracking Based On Histogram Filter And Siamese Network

Jianlong Zhang
Xidian University
Qiao Li
Xidian University
Bin Wang
Xidian University
Chen Chen (cc2000@mail.xidian.edu.cn)
Xidian University
Tianhong Wang
Xidian University
Yang Zhou
The Ministry of water resources of China
Ji Li
Science and Technology on Communication Networks Laboratory

Research Article

Keywords: single object tracking, SiamFC++, Siamese network, dynamic template set, match filter

Posted Date: January 11th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1201475/v1

RESEARCH

Object Relocation Visual Tracking based on Histogram Filter and Siamese Network

Jianlong Zhang[1], Qiao Li[1], Bin Wang[1*], Chen Chen[2*], Tianhong Wang[1], Yang Zhou[3] and Ci He[4]

Abstract

Siamese network based trackers formulate the visual tracking mission as an image matching process by regression and classification branches, which simplifies the network structure and improves tracking accuracy. However, there remain many problems as described below. 1) The lightweight neural networks decreases feature representation ability. The tracker is easy to fail under the disturbing distractors (e.g., deformation and similar objects) or large changes in viewing angle. 2) The tracker cannot adapt to variations of the object. 3) The tracker cannot reposition the object that has failed to track. To address these issues, we first propose a novel match filter arbiter based on the Euclidean distance histogram between the centers of multiple candidate objects to automatically determine whether the tracker fails. Secondly, Hopcroft-Karp algorithm is introduced to select the winners from the dynamic template set through the backtracking process, and object relocation is achieved by comparing the Gradient Magnitude Similarity Deviation between the template and the winners. The experiments show that our method obtains better performance on several tracking benchmarks, i.e., OTB100, VOT2018, GOT-10k and LaSOT, compared with state-of-the-art methods.

Keywords: single object tracking; SiamFC++; Siamese network; dynamic template set; match filter

Introduction

In recent years, visual tracking is a classic research in the computer vision, and also a research hotspot and difficulty. It is a key issue in applications such as surveillance, security, autonomous driving, drones, and smart homes, etc. In spite of the great advances in visual tracking in recent years, it still faces difficulties owing to occlusion, background clutters, scale variation and deformation in dynamic videos.

The traditional trackers, such as the correlation filtering trackers represented by the correlation filter KCF [1] and the core loop architecture tracker CSK [2], have excellent tracking speed, and could quickly update the filter weights online. However, these trackers are unsatisfactory in terms of robustness owing to the impact of the weak semantic information of handcrafted features. As the development of the deep neural networks, C-COT [3] and MDnet [4] improved trackers accuracy by replacing traditional hand-crafted features with deep features. Recently, Siamese network based trackers are gradually being applied to mobile devices owing to their balance between accuracy and efficiency. In the meantime, The development of adversarial training [5] improves the accuracy of Siamese network based trackers and has been applied to fingerprint localization [6], Internet of Things [7–9], intelligent transportation [10] and other fields [11, 12]. These trackers first perform feature extraction using a Siamese network and then exploit a tracking-head network to localize objects from the similarity map. The head network between the search-branch and the template-branch increased the speed and reduces the over-fitting owing to frequent updates of the template. The architecture of these trackers consists of three parts, namely, a Siamese backbone network for template region and search region feature extraction, a similarity matching component for search and template branches information embedding, and a tracking head for information decoding from similarity maps. SiamFC [13] obtained feature through Siamese backbone and introduces a correlation layer to compute the similarity scores of feature maps to localize object with a lightweight architecture that does not need updating the model parameters. SiamFC works efficiently at 86 FPS with high accuracy. RASNet [14] combined a Siamese network with several attention mechanisms

*Correspondence: bwang@xidian.edu.cn; cc2000@mail.xidian.edu.cn
[1]School of Electronic Engineering, Xidian University , Xi'an, China
[2]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China
Full list of author information is available at the end of the article
This article is recommended to submit to our journal by the 2021 IEEE International Conference on Smart Internet of Things (SmartIoT).

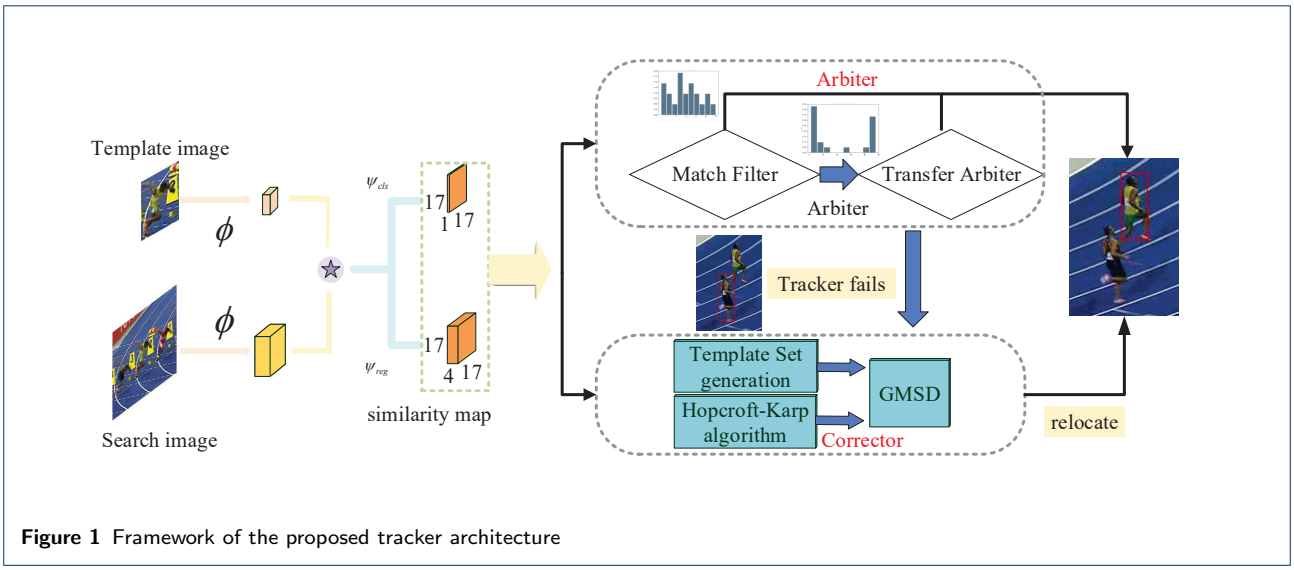

Figure 1 Framework of the proposed tracker architecture

to emphasize more relevant parameters to the object. However, these trackers require a multi-scale transformation to deal with scale variations. In order to get a more accurate and robust result, SiamRPN [15] introduced the RPN [16] into the SiamFC and obtains high accuracy. Both SiamRPN++ [17] and SiamDW [18] reduced the effect of adverse factors (e.g., padding) and decreased the impact of border effects in distinct ways. They introduced deeper neural networks, e.g., ResNet [19] in visual tracking. The anchor-based tracker requires tedious and heuristic configurations, but prior parameters are difficult to fit all objects, which reduce tracker accuracy. Some anchorless trackers, such as SiamFC++ [20], SiamCAR [21], took one or more heads to directly predict the position of the object and regress the bounding boxes from the similarity map.

Siamese network based trackers have made great development, however, the following drawbacks still exist. 1) Owing to the constraints of strict translation invariance and real-time requirement, lightweight neural networks lead to inadequate feature representation. When distractors are presented in the vicinity of the object, it is difficult for the tracker to distinguish which is the right object. 2) Lack of an efficient template update strategy, the single template cannot suit changes in object features, which causes tracking failure when there are large appearance distortions or perspective changes. 3) Due to the lack of an effective arbiter-corrector module, the tracker cannot detect tracking failures, and cannot relocate the object once the object is lost and restart tracking.

SiamFC++

To overcome the above problems, we proposed a SiamFC++ based object relocation tracker. The main contributions of the work are as follows.

- We designed a matching filter arbiter with a hierarchical architecture based on the distance histogram of the candidate objects, which can accurately and quickly find the failure.
- We proposed an efficient corrector that generates a template set by backtracking. The corrector relocates the object by Gradient Magnitude Similarity Deviation (GMSD) and the assignment algorithm measurement to increase the tracker's resistibility to interference.
- Experiments on several challenging benchmarks including VOT-18, GOT-10k, OTB-100 and LaSOT have shown that our proposed tracker is superior to many state-of-the-art trackers.

The reminder of this paper is organized as follows. Section Related Work presents related work on visual tracking. Section Tracker Architecture describes the principles and the implementation of our tracker. Section Experiments provides evaluation and analysis of experimental results. Finally, we summarize our work in Section Conclusion.

Related Work

In this section, we briefly introduce related work about visual tracking.

Early visual tracking methods could be divided two categories according to the tracking mode, namely, the generative model and the discriminative model. With the development of deep learning, visual tracking methods based on deep learning have gradually become mainstream nowadays.

Visual tracking focused on the research of generative model, such as optical flow method [22], particle filter [23] and Meanshift algorithm [24], etc. They firstly established an object model or extracted object

features, and then searched for similar features in subsequent frames. However, the background information of the image is not fully considered. Hence, it is very limited to describe the object through a single mathematical model.

Considering the object and background information at the same time, discriminative model regards the tracking process as a classification or regression problem, and the purpose is to find a discriminant function to separate the object from the background, so as to realize the tracking of the object. The evaluation of the algorithm found that [25], the performance of the tracker could be greatly improved by introducing background information into the tracking model. Therefore, various classifiers were introduced into the visual tracking field. Avidan [26] used support vector machines [27] to distinguish the background and the object, but it is easy to lose the object due to the selected feature is based on a single pixel. TLD [28] used online Ferns [29] to detect objects, while using online random forest algorithm [30] to track objects. In 2010, the cross-correlation was introduced into visual tracking [31]. As a discriminative method, it showed better performance in terms of speed and accuracy. STRCF [32] considered both spatial regularization and time regularization. It could successfully track objects under occlusion and could adapt to larger appearance changes.

The introduction of deep features enhances the feature representation capability of the tracker. HCF [33] utilized the deep and shallow features of the VGG [34] network and incorporated the relevant filters to obtain good tracking performance.

Recently, Siamese network based on trackers have received significant attentions for their balance between high speed and accuracy [13,35]. SINT++ [36] used the positive sample generation network to obtain diverse sample images, by which the robustness of the tracker is improved. SA-Siam [37] utilized two networks to obtain semantic features and appearance features respectively, and introduced the attention mechanism and feature fusion into the semantic branch network. SiamMask [38] solved the problem of visual tracking and object segmentation at the same time, and introduced the segmentation branch to obtain an accurate mask. In this paper, we explored the process of the Siamese network based on trackers, analyzed the correlation between the similarity map and the object, introduced statistical theory to locate the location where the tracker fails and finally relocate the object, The proposed method could improve the performance of the tracker effectively.

Tracker Architecture

As shown in Figure 1, the framework of our tracker consists of SiamFC++, the arbiter and the corrector. Firstly, SiamFC++ produces a similarity map according to the search-branch and the template-branch. Secondly, the arbiter consists of the matching filter arbiter and the transfer arbiter, designed to determine whether the tracking fails. Finally, the corrector implements the repositioning of the tracker, which includes template set update, assignment algorithm and GMSD score.

SiamFC++

SiamFC++ introduces four principles for designing trackers and the anchor-free structure to reduce the prior knowledge, and combines classification and regression branches to increase the tracking accuracy. SiamFC++ extracts the deep feature map of the search regions and the template regions respectively through backbone network, and inputs the feature maps into regression head and classification head respectively to obtain the similarity map, where the highest scoring position represents the object position. The similarity map is the degree of similarity between different positions of the search image and the template. As show in Figure 2 and Figure 3: bounding boxes are the location of the object with different similarity scores. The similarity map has only one center hen there is no distractor round the object(Figure2). When the similarity score of distractor and object is similar, the similarity map has two centers (Figure 3). Because SiamFC++ focuses on both the object and the distractor, the results may shift from objects to distractors. Occlusion, deformation, scale variation

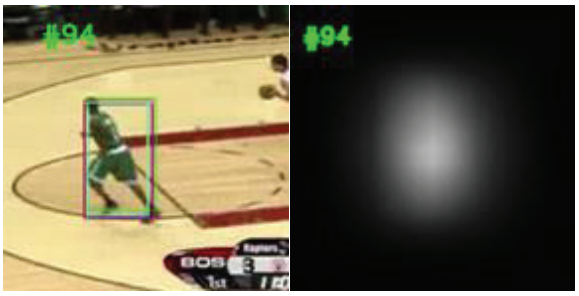

Figure 2 similarity map without distractor

and distractor are the main disruptive factors in tracking datasets. Table 2 shows the percent of SiamFC++ fails factors on VOT-18 dataset. It is clear that 60% of failures are caused by distractor and deformation, therefore, the performance of the tracker could be promoted if we could determine the failures and relo-

Table 1 Notation table

Notation	Meaning	Remarks
N	The length of the video frames	N.A
S_j^i	j-th similarity score of the bounding box in the i-th frame	N.A
R_i	The bounding box set in i-th frame	$R = \{R_1, R_2, \cdots, R_i, \cdots \mid 1 \leq i \leq N\}$
D_i^f	The candidate object set in the f-th frame	$D = \left\{D_1^f, D_2^f, \cdots, D_n^f\right\}$
D_i	i-th candidate object	$D = \{D_1, D_2, \cdots, D_n\}$
T_p	p-th template	$T = \{T_1, T_2, \cdots, T_{p-1}, T_p\}$
J_i	The set of tracking failure frames	$J \in \{J_1, J_2, \cdots, J_m\}$
n	The number of candidate objects	N.A
q	The number of winners	N.A
O_i	i-th center of the object	$O_i = \{x_i, y_i\}$
H_i	The frequency of the i-th bin in the histogram	$H = \{H_1, H_2, \cdots, H_k\}$

Figure 3 similarity map with distractor

Table 2 challenge factors

Challenge	ratio
distractor	0.3
deformation	0.3
scale variation	0.1
occlusion	0.1
other	0.2

cate the object. For this purpose, we designed a system that contains two modules, namely, the arbiter and the corrector. In order to explain its mathematical principle, we let N be the length of the video frames. S_j^i the similarity score of the bounding box with the similarity rank j in the i-th frame. $R = \{R_1, R_2, \cdots, R_i, \cdots \mid 1 \leq i \leq N\}$ is the bounding box set of the object. $D = \left\{D_1^f, D_2^f, \cdots, D_n^f \mid S_1^f > S_2^f > \cdots > S_n^f\right\}$ is the candidate object set in the f-th frame. $T = \{T_1, T_2, \cdots, T_{p-1}, T_p\}$ is the templates set, where T_p is the p-th template; $J \in \{J_1, J_2, \cdots, J_m\}$ is the set of tracking failure frames, which indicates that the tracker fails in the J_i frame.For the convenience of reading, we list the symbols used in this paper in Table 1.

Arbiter

Herein, the purpose of the arbiter is to determine if the tracker fails. We need to reposition the object once the tracking fails. The main matching filter arbiter based on the candidate object histogram determines whether distractors exist. The transfer arbiter determines whether the object is transferred based on the change of the object in relative position, and finally both together determine if the tracker is not working.

Match Filter

Considering the above observation that the similarity map shows two highlight areas corresponding to the distractor and the object, respectively, when the distractor is presented, we attempt to arbitrate the existence of the distractor using the histogram of the distance between the region centroids. Let $L_{ij} = \sqrt{(y_i - y_j)^2 + (x_i - x_j)^2}$ be the Euclidean distance between the object centers where (x_i, y_i) is the center of D_i. The distance is small among similar candidate objects, while is large between different types. Therefore, the distance shows a trend of bipolar distribution. The histogram H is similar to a band stop filter, as shown in Figure 5. Equation 1 is a histogram, i.e.,

$$H = \{H_1, H_2, \cdots, H_k\}, \tag{1}$$

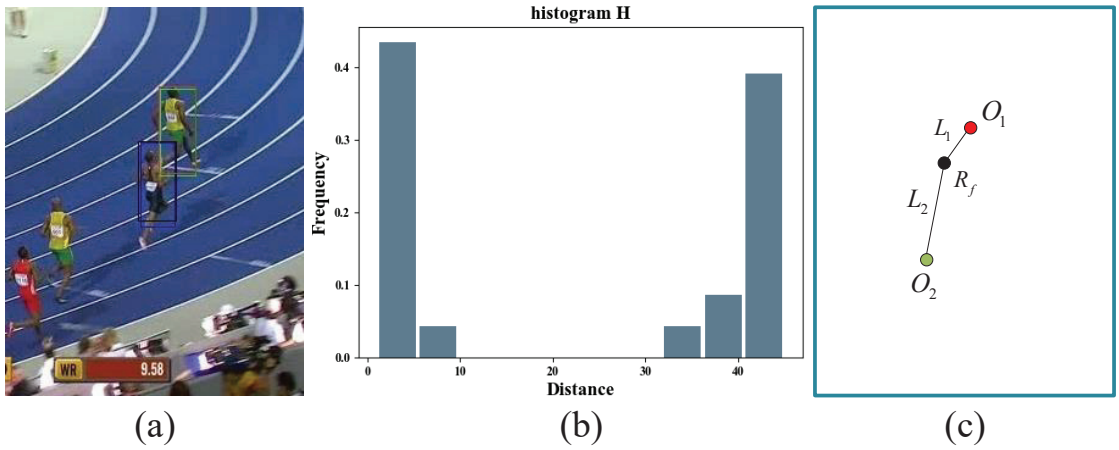

Figure 4 Transfer arbiter

where H_i is the frequency of the i-th bin, k is the number of histogram bins ($k >= 10$), the total frequency is C_k^2. We input H into the band-stop filter F to obtain the filter output Z. Equation 2 is a discrete band stop filter. The calculation of Z is shown in equations 3. When the output is greater than the threshold T, it indicates that there are interferences in the image.

$$F = \{F_1, F_2, \cdots, F_k\}, \tag{2}$$

$$Z = \sum_{i=1}^{k} H_i \times F_i, \tag{3}$$

Transfer Arbiter

The matching filter arbiter could determine whether the distractor exists, but it could not determine whether the object is displaced by the distractor. As shown in Figure 4(a), the red border is the object position, but the distance histogram also takes the shape of a band stop filter, as shown in Figure 4(b). Therefore, it is difficult to determine if the tracker fails if only using the matching filter arbiter. It is known that the distance among the objects in adjoining frames changes less than the distance between the distractor and the object. The relative distance changes when the tracking result is moved to the distractor. It is possible to further determine whether the tracker fails based on the changes in the relative locations of the distractor and the object in the adjacent frames.

The framework of the transfer arbiter is shown in Figure 6: When the $f+1$ frame passes through the matching filter arbiter, is classified via K-means algorithm, where K=2. As shown in Figure 4(c),

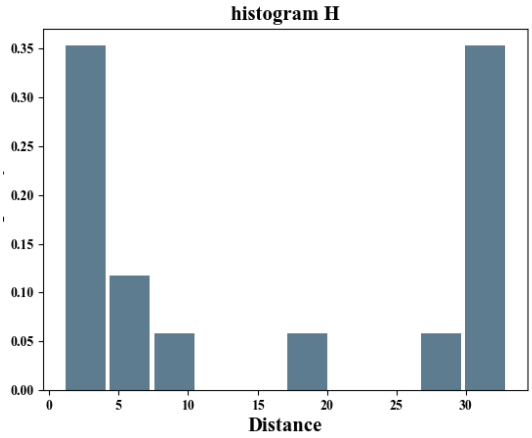

Figure 5 Histogram H

the center closer to (x_1, y_1) is the object position O_1 , and the other center is the distractor position O_2. $L_1 = \sqrt{\left(x_{R_f} - x_{O_1}\right)^2 + \left(y_{R_f} - y_{O_1}\right)^2}$ is the Euclidean distance between R_f and O_1. $L_2 = \sqrt{\left(x_{R_f} - x_{O_2}\right)^2 + \left(y_{R_f} - y_{O_2}\right)^2}$ represents the distance between R_f and O_2. The procedure of transfer arbiter is as follows. In order to demonstrate the effectiveness of the proposed arbiter, we calculate Track-

Algorithm 1 Transfer Arbiter.

```
if L1 < L2 then
    Object is not transferred and tracker work;
else
    Tracker fails;
end if
```

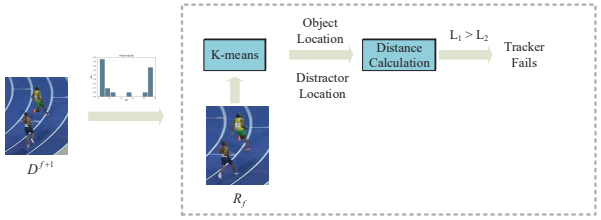

Figure 6 Transfer arbiter framework

ing Invalid Accuracy Ratio (TIAR) on every dataset, which is the percentage of correct judgments among tracking failure sequences. Table 3 shows that arbiter can detect most of the tracking failures. Although the length of the GOT-10k video sequence is long, it does not perform satisfactorily, the arbiter still works in 40% of the failure scenarios.

Table 3 TIAR of the arbiter

Dataset	TIAR
OTB-15	0.68
VOT-18	0.59
LaSOT	0.82
GOT-10k	0.40

Corrector

The object needs to be relocated once the tracker fails. We proposed a corrector consisting of dynamic template update, assignment algorithm and GMSD. Updated by the similarity backtracking, the template set is to find the previous tracking results. The assignment algorithm is used for selecting the winner set which is similar to object. Finally, the object is relocated by computing the GMSD between the template set and winner set.

Dynamic Template Set

In long-term tracking, a single template cannot handle the changes of object appearance, such as, 1) When the object appearance changes gradually, the error accumulates and finally the object cannot match the template well; 2) When the object appearance changes suddenly and drastically, the object is very different from the template. Therefore, we proposed a template update procedure which automatically adds the result different from the template into the template set, so that the diversity of the template set could be enriched. The template set update algorithm is as follows. The process of generating the template set is shown in Figure 7. The template image is shown in (a). When the appearance of the object in (b) changes significantly, the tracker is disabled. Meanwhile, the object in frame 151 in (c) is added to the template set by Algorithmg 2. The template will be more similar to the object in the next frames.

Algorithm 2 The template set update algorithm.

while tracker fails **do**
  if fails at J_i **then**
    Similarity backtracking:
    $S_1^z = \min\left(S_1^{J_{i-1}}, S_1^{J_{i-1}+1}, \cdots, S_1^{J_i}\right)$;
    Add R_z to M
  end if
end while

(a) Template (b) Tracker fails (c)New Template

Figure 7 Dynamic Template set update

Assignment Algorithm

Since there are lots of candidate objects, the corrector utilizes the Hopcroft-Karp algorithm [39] to select the set of winners with high similarity from the candidate objects and calculates the GMSD between the winners and the template to reduce the computation and improve the speed.

Hopcroft-Karp algorithm is to realize bipartite graph matching. Compared with the Kuhn-Munkres [40] algorithm, it looks for multiple augmentation paths at a time. This can further decrease the time complexity and get the optimal complete match. The bipartite graph matching model is shown in Figure 8. The matching process is as follows.

1) Take an initial match M from $\mathrm{G} = (X, Y; w)$. The weight w calculation between different vertex is shown in equation 4;
2) While there exists an augmenting path P, remove matching edges of P from M and add not-matching edges of P to M (This increases size of M by 1 as P starts and ends with a free vertex, i.e. a node that is not part of matching.) ;
3) Retuen M.

$$w\left(D_i^f, D_j^{f+1}\right) = \phi\left(\mathrm{D}_i^f\right) \otimes \phi\left(\mathrm{D}_j^{f+1}\right) \quad (4)$$

where $\phi(.)$ is the Siamese backbone for feature extraction, and $\otimes$ is the cross-correlation operator.

We obtained a complete match of D^f, D^{f+1} by Hopcroft-Karp algorithm. Define $\mathrm{C} = \left\{C_1^{f+1}, \cdots, C_q^{f+1}\right\}$ as the winner set, C_i^{f+1} is the candidate object that D_i^f matches, and q is the number of winners.

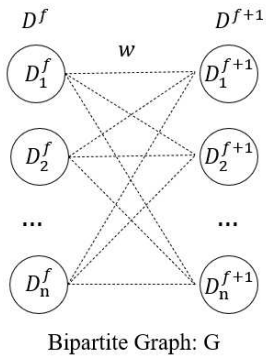

Figure 8 Bipartite Graph Matching Model

GMSD Relocation

Since the backbone of SiamFC++ has difficulty in distinguishing the distractor from the object, it is essential to choose another efficient algorithm to restart tracker. We introduce the Gradient Magnitude Similarity Deviation (GMSD) to relocate the object. GMSD can distinguish the object in the appearance and structure, and only uses the gradient magnitude as a feature can generate a high-accuracy score. It can precisely locate the objects that are similar to the template in the case of similar semantic information. The calculation of GMSD is shown in equations 5, 6, 7 and 8 as follows.

$$h_x = \begin{bmatrix} 1/3 & 0 & -1/3 \\ 1/3 & 0 & -1/3 \\ 1/3 & 0 & -1/3 \end{bmatrix} \quad h_y = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 \\ -1/3 & -1/3 & -1/3 \end{bmatrix} \tag{5}$$

$$m_r(i) = \sqrt{(r \otimes h_x)^2(i) + (r \otimes h_y)^2(i)}, \tag{6}$$

$$m_d(i) = \sqrt{(d \otimes h_x)^2(i) + (d \otimes h_y)^2(i)}, \tag{7}$$

$$GMS(i) = \frac{2m_r(i)m_d(i) + c}{m^2(i) + m_d^2(i) + c}, \tag{8}$$

where h_x, h_y are the Prewitt operator used to calculate the image gradient. $m_r(i)$ and $m_d(i)$ are the image gradient. c is a small constant. When $f+1$ frame tracking fails, the GMSD measurement between the template set and the winner set is calculated to get the object position, the associated formula is

$$\mathrm{S}_0 = \max\left(\underset{1\le i\le q, 1\le j\le p}{GMS}(\mathrm{C}_i,\ \mathrm{T}_j)\right), \mathrm{C}_i \in C, T_j \in T, \tag{9}$$

Experiments

We used GoogLeNet as the backbone network of SiamFC++. The number of candidate objects $n = 10$. The number of winners $q = 5$, filter center frequency $f_0 = 5$, and the band stop width = 8. Our tracker is realized with PyTorch on a PC with Nvidia GTX 2080Ti, Intel(R) Core (TM) i7-7820X CPU @ 3.60GHz.

Result on Several Benchmarks

We compared our proposed tracker with some of state-of-the-art trackers on four tracking benchmarks as shown in Table 4 and Figure 9. Our tracker obtains state-of-the-art performance.

OTB2015 Benchmark OTB2015 includes 100 videos, which provides a standard evaluation benchmark for trackers. A comparison with state-of-the art trackers is shown in Figure 10 in terms of success plots of OPE. Our tracker obtained a success score of 0.727, which achieved state-of-the-art performance.

VOT Benchmark VOT2018 consists 60 videos with challenging factors such as deformation, occlusion, etc. The performance of the tracker is evaluated by the accuracy.As shown in Figure 11, the bounding boxes are the candidate object with higher similarity score and the red bounding box has the highest similarity score, which is the tracking result. In Figure 11(a), when there is no distractor in the frame and the object deformation is small, the bounding boxes with high scores are concentrated on one object, and the tracking is successful at this time. When there is a distractor in the image and the tracker fails, as shown in Figure 11(b), the tracking result has changed from Bolt to another athlete. At this time, some bounding box is still positioned on the Bolt. By comparing the GMSD between the candidate objects and the template, the object could be found again and the tracker could be corrected. After removing the reinitialize mechanism, our tracker has the highest accuracy and obtains an accuracy score of 0.533 due to we could relocate the object, which is a good improvement compared to SiamFC++.

Table 4 Results on several benchmarks

Tracker	SiamFC	ECO	SiamRPN++	ATMO	SiamFC++	Ours
OTB-15 Success	58.2	70.0	69.6	66.9	68.3	72.7
VOT-18 Accuracy[1]	0.412	0.404	0.484	0.478	0.480	0.533
LaSOT Success	33.6	32.4	49.6	51.5	54.4	57.5
GOT-10k AO	34.8	31.6	51.8	55.6	59.5	61.2

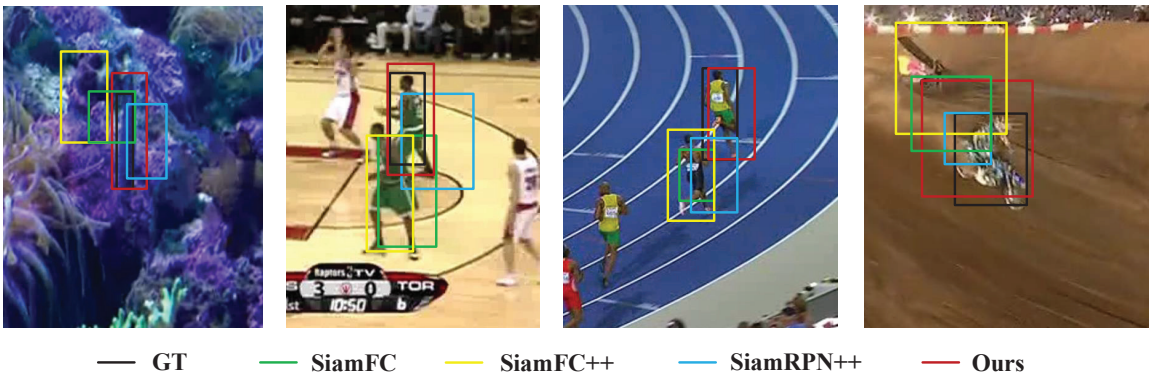

Figure 9 Qualitative comparison results

LaSOT Benchmark LaSOT contains 1400 videos, which is a high-quality, large-scale datasets for long-term tracking. By relocating the lost objects, the tracker reached the top performance with a success score of 0.575, which shows that our tracker also has a good performance in difficult scenarios.

GOT-10k Benchmark GOT-10k contains a lot of small objects and they become small as objects move and the viewpoint changes. It causes our tracker not to capture the real object when the tracker fails, which keeps the tracker from achieving a better performance.

Conclusion

In this paper, we proposed a Siamese network based on trackers with a generic arbiter-corrector module. It could resolve the tracking failure problem caused by appearance changes of object and distractors. The arbiter proposed an efficient architecture based on the match filter that determines whether the tracker has lost the object. The template is updated to increase the tracker's resistance to interference. The corrector repositions the object by GMSD and dynamic template set. The generic arbiter-corrector module can be easily integrated in other trackers. The associated experiments show that the proposed arbiter-corrector mechanism is effective in improving the accuracy of the tracker. Mining the association of template sets using online learning to improve the robustness of tracking is the next step of the research.

[1]VOT-18 will restart the tracker when the tracker fails. In order to validate the arbiter-corrector system, the reinitialize mechanism is removed.

Abbreviations
Not applicable.

Acknowledgements
We are thankful to Electronic Engineering of Xidian University for providing an environment for editing manuscripts and experiments.

Authors' contributions
The authors had worked equally during all this paper's stages. All authors read and approved the final manuscript.

Funding
This work was supported by the National Key Research and Development Program of China(2020YFB1807500), the Aeronautical Science Foundation of China (2018ZC81001), the National Natural Science Foundation of China (61902292, 62072360, 61971331, 62001357), the Key Research and Development Plan of Shaanxi province (2019ZDLGY13-07, 2021ZDLGY02-09, 2020JQ-844, 2019ZDLGY13-04), the Key Laboratory of

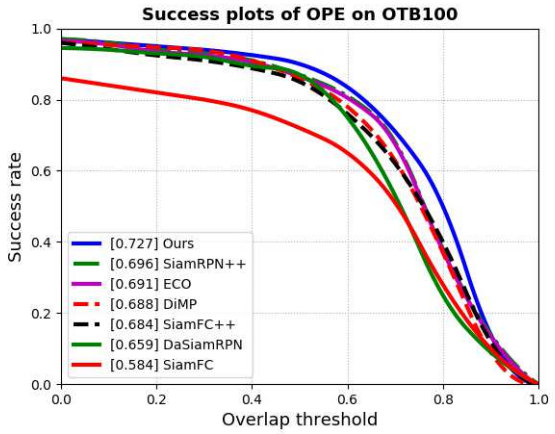

Figure 10 Success plots on OTB100

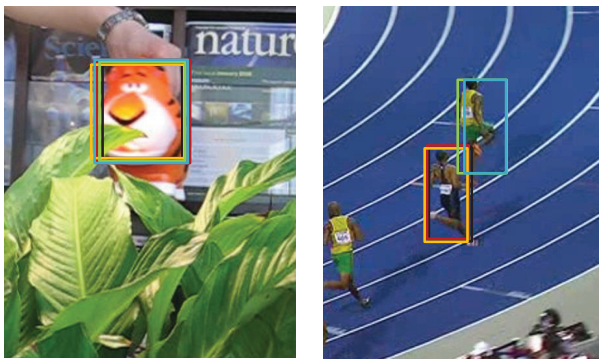

Figure 11 Candidate objects and correction process

Embedded System and Service Computing (Tongji University) (ESSCKF2019-05), Ministry of Education, the Xi'an Science and Technology Plan (20RGZN0005) and the Xi'an Key Laboratory of Mobile Edge Computing and Security (201805052-ZD3CG36).

Availability of data and materials

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]School of Electronic Engineering, Xidian University , Xi'an, China. [2]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China. [3]The Ministry of water resources of China , Beijing, China. [4]Science and Technology on Communication Networks Laboratory, ShijiaZhuang, China.

References

1. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE transactions on pattern analysis and machine intelligence **37**(3), 583–596 (2014)
2. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: European Conference on Computer Vision, pp. 702–715 (2012). Springer
3. Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M.: Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: European Conference on Computer Vision, pp. 472–488 (2016). Springer
4. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4293–4302 (2016)
5. Lv, N., Ma, H., Chen, C., Pei, Q., Zhou, Y., Xiao, F., Li, J.: Remote sensing data augmentation through adversarial training. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **14**, 9318–9333 (2021)
6. Wu, C., Qiu, T., Zhang, C., Qu, W., Wu, D.O.: Ensemble strategy utilizing a broad learning system for indoor fingerprint localization. IEEE Internet of Things Journal (2021)
7. Qiu, T., Lu, Z., Li, K., Xue, G., Wu, D.O.: An adaptive robustness evolution algorithm with self-competition for scale-free internet of things. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, pp. 2106–2115 (2020). IEEE
8. Qiu, T., Zhang, S., Si, W., Cao, Q., Atiquzzaman, M.: A 3d topology evolution scheme with self-adaption for industrial internet of things. IEEE Internet of Things Journal (2020)
9. Cong, W., Chen, C., Qingqi, P., Zhiyuan, J., Shugong, X.: An information centric in-network caching scheme for 5g-enabled internet of connected vehicles. IEEE TRANSACTIONS ON MOBILE COMPUTING (2021). doi:10.1109/TMC.2021.3137219
10. Chen, C., Jiange, J., Rufei, F., Lanlan, C., Cong, L., Shaohua, W.: An intelligent caching strategy considering time-space characteristics in vehicular nameddata networks. IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (2021). doi:10.1109/TITS.2021.3128012
11. Fu, S., Atiquzzaman, M., Ma, L., Lee, Y.-J.: Signaling cost and performance of sigma: A seamless handover scheme for data networks. Wireless Communications and Mobile Computing **5**(7), 825–845 (2005)
12. Zhang, J., Li, Q., Wang, B., Chen, C., Wang, T., Zhou, Y., Li, J.: Object relocation visual tracking based on siamese network. In: 2021 IEEE International Conference on Smart Internet of Things (SmartIoT), pp. 95–100 (2021). IEEE
13. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: European Conference on Computer Vision, pp. 850–865 (2016). Springer
14. Wang, Q., Teng, Z., Xing, J., Gao, J., Hu, W., Maybank, S.: Learning attentions: residual attentional siamese network for high performance online visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4854–4863 (2018)
15. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8971–8980 (2018)
16. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28**, 91–99 (2015)
17. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4282–4291 (2019)
18. Zhang, Z., Peng, H.: Deeper and wider siamese networks for real-time visual tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4591–4600 (2019)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
20. Xu, Y., Wang, Z., Li, Z., Yuan, Y., Yu, G.: Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 12549–12556 (2020)
21. Guo, D., Wang, J., Cui, Y., Wang, Z., Chen, S.: Siamcar: Siamese fully convolutional classification and regression for visual tracking. In:

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6269–6277 (2020)
22. Lucas, B.D., Kanade, T., *et al.*: An iterative image registration technique with an application to stereo vision. (1981). Vancouver, British Columbia
23. Nummiaro, K., Koller-Meier, E., Van Gool, L.: An adaptive color-based particle filter. Image and vision computing **21**(1), 99–110 (2003)
24. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on pattern analysis and machine intelligence **24**(5), 603–619 (2002)
25. Collins, R., Zhou, X., Teh, S.K.: An open source tracking testbed and evaluation web site. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, vol. 2, p. 35 (2005)
26. Avidan, S.: Ensemble tracking. IEEE transactions on pattern analysis and machine intelligence **29**(2), 261–271 (2007)
27. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural processing letters **9**(3), 293–300 (1999)
28. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE transactions on pattern analysis and machine intelligence **34**(7), 1409–1422 (2011)
29. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: 2007 IEEE 11th International Conference on Computer Vision, pp. 1–8 (2007). Ieee
30. Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P., Feuston, B.P.: Random forest: a classification and regression tool for compound classification and qsar modeling. Journal of chemical information and computer sciences **43**(6), 1947–1958 (2003)
31. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2544–2550 (2010). IEEE
32. Li, F., Tian, C., Zuo, W., Zhang, L., Yang, M.-H.: Learning spatial-temporal regularized correlation filters for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4904–4913 (2018)
33. Ma, C., Huang, J.-B., Yang, X., Yang, M.-H.: Hierarchical convolutional features for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3074–3082 (2015)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
35. Tao, R., Gavves, E., Smeulders, A.W.: Siamese instance search for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1420–1429 (2016)
36. Wang, X., Li, C., Luo, B., Tang, J.: Sint++: Robust visual tracking via adversarial positive instance generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4864–4873 (2018)
37. He, A., Luo, C., Tian, X., Zeng, W.: A twofold siamese network for real-time object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4834–4843 (2018)
38. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: A unifying approach. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1328–1338 (2019)
39. Hopcroft, J.E., Karp, R.M.: An n^5/2 algorithm for maximum matchings in bipartite graphs. SIAM Journal on computing **2**(4), 225–231 (1973)
40. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics (NRL) **52**(1), 7–21 (2005)