17th International Conference on Machine Vision Applications (MVA)
Fully Online, July 25–27, 2021.

Augmenting Discriminative Correlation Filters with Stereo Blob Tracking for Long-Term Tracking of Underwater Animals

Miao Zhang, Stephen Rock
Stanford University
496 Lomita Mall, Stanford, CA, USA 94305
{miaoz2, rock}@stanford.edu

Abstract

This paper presents a vision-based model-free long-term tracking algorithm to be used on-board autonomous underwater vehicles (AUVs) for long duration marine animal observation missions. During underwater tracking missions, drifting and losing track of targets after they leave the field of view are two major problems with state-of-the-art tracking algorithms. To achieve the long-term tracking goal, the proposed method gained drift resistance and target re-capturing ability by combining the merits of two mature short-term trackers: stereo blob tracking and discriminative correlation filter (DCF). In our approach, stereo blob tracking acts as complementary supervision to correct drift and to guide DCF to learn target appearances online before any tracking interruptions. The target information learned is then used to help re-capture the target after a tracking failure. In our experiments on field data, compared to running DCF alone, running the proposed augmented tracker increased average bounding box accuracy by 45% and eliminated drift-caused tracking failures. Our tracking algorithm also achieved 86% target re-capturing success.

1 Introduction

1.1 Motivation

Marine biologists primarily rely on short-range videos collected by underwater vehicles to study the behavior of gelatinous ocean animals. Most of these data are currently collected using human-piloted remotely operated vehicles (ROVs) equipped with science cameras to film the animals in situ. However, since observing time is limited for human piloted missions, prolonged tracking and filming for more than 24 hours requires an autonomous underwater vehicle (AUV) equipped with a position-based visual servoing system that uses vision algorithms to recognize and localize targets in camera images. Additionally, for long-term tracking, the vision algorithm needs to be able to re-identify targets after temporarily losing them. The challenges in target recognition come from the many possible target appearances that result from deformation and change of perspective. Offline machine learning classification models [1] are showing great promise for discerning the target; however, their performance is limited to targets in the existing data base. A model-free tracker that learns the target appearance online offers flexibility in tracking additional targets of opportunity.

1.2 Related Work

Many tracking algorithms have demonstrated good short-term performance when tracking deformable targets. One of them is 3D stereo tracking based on blob detection [2,3], which has been used on tracking underwater animals for decades [1,4,5]. Stereo blob tracking works well when the target is consistently in view because it does not drift, and it can adapt to a wide range of size and appearance variations. However, under disruptive circumstances such as when a target is occluded or temporarily leaves the field of view, the blob tracking algorithm has limited ability to re-identify the target. Some efforts have been made to assist the target identification process, such as training a primitive target feature vector [6], taking advantage of stereo separation during occlusion, and propagating the last seen world-frame position of the target. Such practices demonstrated success in tracking a single siphonophore fully autonomously for five hours [1].

Another algorithm that has performed well in short-term tracking of deformable targets is the discriminative correlation filter (DCF) [7–10]. DCF is one of the two dominating short-term tracking branches in current visual tracking literature, occupying 68% of submissions in the 2020 Visual Object Tracking challenge [11]. Compared to SiameseNet-based methods [12] (the other popular branch) that generally require GPUs with large memory, most DCF-based methods run real-time using a regular CPU, which is desirable for underwater tracking missions with limited on-board computational resources. In addition, frequent updates offer DCF trackers some adaptability to a target's appearance change. However, monocular DCF trackers are known to drift under illumination changes, target deformation, and various other factors. The drift often deteriorates for long duration tracking. DCF trackers are also subject to limited target recovery capability after disruptive events.

There are many model-free long-term tracking algorithms, including TLD [13] from a decade ago and the newly introduced SiameseNet-based long-term trackers [14]. Moreover, authors in [15–17] showed that long-term tracking can be achieved by augmenting DCF with an extra online-trained appearance model for re-detection. Particularly, LCT [15] trained a random-fern classifier; LCMHT [16] trained an SVM classifier; MUSTer [17] kept a long-term memory that supported SIFT keypoint matching and RANSAC estimation. More recently, FuCoLoT [18] proposed to use DCF for both tracking and re-detection, removing the necessity of training an extra classifier. However, long-term tracking of deformable targets remains a challenge, and the potential of correlation filters (CFs) in target re-identification is still relatively under-explored.

1.3 Contribution

This paper presents a vision-based approach that demonstrates preliminary success on field data for long duration tracking of underwater animals. (1) We introduce a method to integrate stereo blob tracking and DCF tracking that exploits the stereo blob tracker's reliability in the absence of interruptions and the under-explored potential of CFs in target re-identification. (2) We propose a target re-identification logic based on template matching which uses CFs. The identification process utilizes proposals from the stereo blob tracker to focus queries on salient regions, instead of performing an image-wide search and detection as in [18]. Moreover, unlike traditionally measuring the similarity between the features of the template image and the query image, we used a score metric that measures the similarity between the two response maps that result from performing correlation operations on each of the two images with the same CF. This score metric, inspired by [19], conveniently reuses mechanically-produced CFs for re-identification, and thus avoids using an extra feature encoder. (3) Rather than updating the CFs at a set of fixed frequencies as in [18], the proposed approach constructs a long-term memory of target appearance by selectively storing CF templates, which is especially useful for targets with large appearance variation. The process of storing templates is also supervised by the stereo blob tracker to ensure location accuracy, thus effectively mitigating the drift problem in DCF tracking.

2 The Augmented Tracker

The presented long-term tracker: Augmented Channel and Spatial Reliability Discriminative Correaltion Filter (ACSRDCF) uses CSR-DCF [9] as the base short-term DCF tracker and fuses it with a stereo blob tracker similar to the one in [1]. ACSRDCF adds a long-term memory component and a mode switching functionality between normal tracking and active search. Fig. 1 summarizes the tracking logic of ACSRDCF.

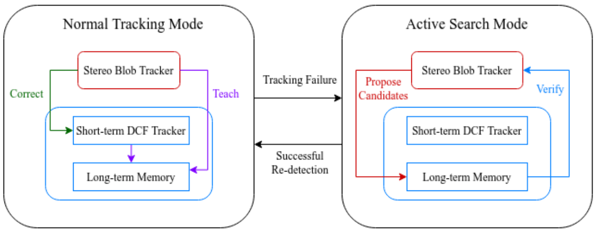

Figure 1. ACSRDCF high-level tracking logic

2.1 Normal tracking mode

After initialization, normal tracking is the default mode in which the stereo blob tracker and the DCF tracker run in parallel. At every frame, the DCF tracker updates the CF while the blob tracker makes stereo and temporal associations using the Hungarian algorithm. In addition, the long-term memory $\mathcal{H}$ is generated by storing CF templates that are reliable and unique. This is accomplished in two steps:

Check reliability and make correction: The stereo blob tracking result is accepted as a reliable reference when the blob association cost is low and tracking interruptions are absent. The DCF tracker's bounding box is corrected to match the stereo blob tracker's bounding box when the latter is reliable and the intersection over union (IoU) between the two bounding boxes falls below a given threshold. The IoU for two bounding boxes A and B is defined as $IoU(A, B) = \frac{area(A \cap B)}{area(A \cup B)}$.

Check redundancy and memorize: Memorizing the current template is considered necessary (not redundant) when all templates currently in $\mathcal{H}$ fail to identify the current reliably tracked target. When necessary, $\mathcal{H}$ memorizes a new template, which contains the current CF, $f_{hist}(\mathbf{r})$ (the normalized histogram of the response map $\mathbf{r}$ that results from correlating the CF with the image), and associated stereo tracking information.

2.2 Active search mode

The proposed tracker transitions from normal tracking mode to active search mode when a tracking failure is detected under interruptions. At any frame in the search mode, blob detection provides candidates to be identified. First, blob candidates that fall outside of a given size range are eliminated. Next, detection confidence scores are computed for all blob candidates using the memories learned in the normal tracking mode. Last, the algorithm makes decisions on whether to re-initialize tracking or to continue searching based on computed scores. Specifically:

Detect tracking failure: The proposed tracker enters uncertain status when the blob association cost rises above a given threshold and the Peak-to-Sidelobe Ratio (PSR) [18,19] of the response map $\mathbf{r}$ falls below the adaptive threshold computed from the average PSR of recent frames. Tracking failure is detected after persistent uncertain status.

Eliminate by size: The true target size range is proportional to the pixel width w and height h in the image and inversely proportional to the orthogonal distance d between the camera plane and the target. We only keep blobs that meet $0.8 < \frac{1}{2}(\frac{w_{det}}{w_{temp}} + \frac{h_{det}}{h_{temp}})\frac{d_{det}}{d_{temp}} < 1.2$ with any template in $\mathcal{H}$, where subscript det, $temp$ denote "detection" and "template", respectively.

Compute detection confidence score: Ideally, when the query image is similar to the template image, the distribution of the response of the two images correlating with the same CF should also be similar. The Jensen Shannon Divergence (JSD) can be used to measure similarity between $f_{hist}(\mathbf{r}_{det})$ (the normalized histogram of the query response map) and $f_{hist}(\mathbf{r}_{temp})$ (the normalized histogram of the template response map), and this measurement can then be used to infer similarity between the query image and the template image [19]. A lower $JSD(f_{hist}(\mathbf{r}_{det})||f_{hist}(\mathbf{r}_{temp}))$ indicates a better query-template match. Meanwhile, high $max(\mathbf{r}_{det})$ and high $PSR_{\mathbf{r}_{det}}$ often indicate a higher detection confidence [18]. The overall confidence score of the detection i matching template j is proposed as: $S_{ij} = PSR_{\mathbf{r}_i} \cdot exp(-k \cdot JSD(f_{hist}(\mathbf{r}_i)||f_{hist}(\mathbf{r}_j))) \cdot max(\mathbf{r}_i)$, where k is a scalar constant that scales the importance of JSD in S. The detection confidence score for blob i is $max(S_{ij}), \forall j$.

Make decision: If the highest detection confidence score among the blob candidates exceeds a given threshold, normal tracking re-initializes at the blob with the highest score; otherwise the search continues.

3 Experiments

3.1 Data and hardware

The performance of the proposed algorithm is tested on data previously collected by an ROV in a five-hour semi-autonomous ocean dive. We extracted eight video clips that each continually followed a target for at least one minute. The true target locations were manually annotated once every 100 frames. These annotated frames act as anchor checkpoints for tracking evaluation. The extracted data contain 61680 images (or 35 min of video). The data cover five tracked targets and numerous encountered species while following the main target. Furthermore, to demonstrate the ability of the proposed algorithm to recover lock on a lost target, three videos were generated by partially cropping the original camera view to simulate scenarios where the animal leaves the field of view and later returns. We implemented the proposed ACSRDCF tracker in MATLAB, and performed all the experiments on an Intel i7-9750H CPU (2.6 GHz) with 16 GB RAM.

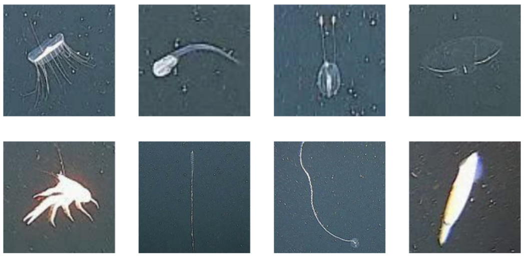

Figure 2. Underwater animals in current data. First five: tracked targets; Last three: examples of other species encountered during tracking

3.2 Evaluation of performance agianst drift

We compared ACSRDCF with CSR-DCF [9], which ACSRDCF is based on, and FuCoLoT [18], which is a top-performing long-term DCF tracker according to [20], using their publicly available source code. We ran all three algorithms on eight long duration tracking videos. At every annotated anchor frame, we compared each tracker's IoU with the ground truth which we later referred to as the accuracy. A tracker was considered to fail when the accuracy fell below 0.2, then the failed tracker would be reinitialized. No failure was observed for ACSRDCF, compared to 19 and 20 failures for CSR-DCF and FuCoLoT respectively. The ACSRDCF accuracy averaged over 0.9 while the accuracy of CSR-DCF and FuCoLoT averaged around 0.45. To visually compare robustness, we adapted the robustness indicator $e^{-L^{F_{0.2}}/N}$ introduced in [21], which can be interpreted as the probability of the tracker continuously tracking the target for over L checkpoints. N denotes the video sequence length and $F_{0.2}$ is the number of failures. As shown in Fig. 3, where one data point marks the performance on one video clip, the proposed tracker performed considerably better in terms of both accuracy and robustness on all eight video clips, indicating significant drift mitigation in long duration tracking.

3.3 Evaluation of re-detection performance

First, the rate of correct target recognition on annotated images using long-term memory built with different amount of training information was recorded in Table 1. As expected, the overall recognition rate

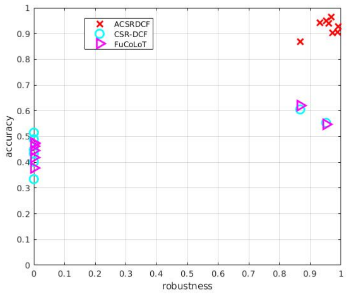

Figure 3. Accuracy vs. robustness visualization. An ideal performance should have both accuracy and robustness close to 1.

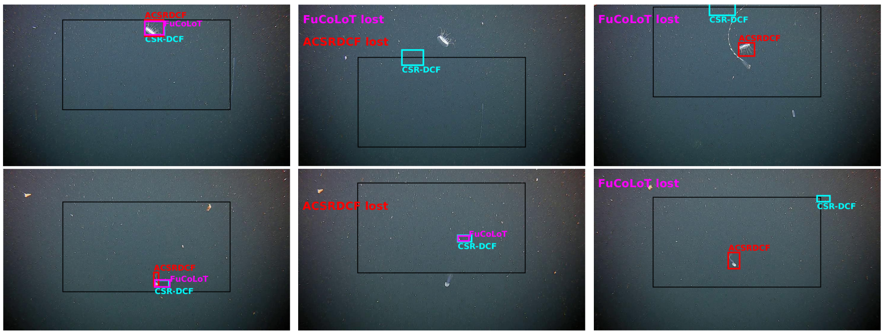

Figure 4. Screenshots of successful tracking recovery of ACSRDCF. Each row shows screenshots before the target goes out of view (left), when the target is out of view (middle), and when the target comes back in view (right). Black box: cropped field of view. Red: ACSRDCF result. Cyan: CSR-DCF result. Magenta: FuCoLoT result. In these two samples where ACSRDCF demonstrated recovery success, CSR-DCF showed no recovery ability, and FuCoLoT reported false positives and false negatives.

Table 1. Correct target recognition rate after different amount of observation time

Recognition	Portion of video observed		
rate	33.3%	66.7%	100%
range	[.28 .88]	[.30 .88]	[.65 .9]
mean±std.	.52±.22	.64±.20	.80±.12

Table 2. Long-term tracking performance evaluation on out of view challenge

	Re-detection	Pr	Re	F	FPS
CSR-DCF	0/35	0.02	0.03	0.02	8.7
FuCoLoT	23/35	0.80	0.66	0.73	5.4
ACSRDCF	30/35	0.99	0.73	0.84	5.2
ACSRDCF (oracle)	35/35	0.99	0.97	0.98	5.2

increases with more training time. However, the rate of increase is closely associated with the diversity of target profiles seen during training. In cases when the target remains still, more observing time does not yield more information.

Second, we compared the target re-acquisition performance of ACSRDCF with CSR-DCF [9] and FuCoLoT [18] on targets being temporarily out of view. All algorithms were evaluated on the three generated videos with cropped fields of view. Besides running ACSRDCF regularly by starting with an empty memory, we also ran the same ACSRDCF algorithm but starting with a target memory previously learned from the whole video. The latter run is called ACSRDCF(oracle), which serves as an upper bound for ACSRDCF performance when given adequate chance to observe the target. Table 2 records the number of successful re-detection, as well as precision, recall, F-score defined in [20], and running speed in Frames-Per-Second (FPS). Our tests suggest that ACSRDCF obtains the ability to recover tracking after an observation time as short as three seconds. Running ACSRDCF with evolving memory achieved 86% re-detection success, which kept up with the success rate of running ACSRDCF(oracle) (100%). As shown in Table 2, ACSRDCF performed better in all listed metrics compared to CSR-DCF and FuCoLoT, while only running marginally slower. Fig.4 shows samples of successful recoveries when running ACSRDCF on two different animals.

4 Conclusion

We proposed a model-free long-term single object tracker that gained its long-term tracking ability by integrating two mature short-term trackers. Particularly, the target re-detection functionality is achieved by building a long-term correlation filter memory using the stereo blob tracker as supervision during steady tracking. On field data, the proposed method demonstrated improved robustness against drift and promising target re-detection success against out of view events. These findings suggest potential of the proposed method in the application of long duration tracking of underwater animals. However, the proposed algorithm's performance against other challenging scenarios, such as full occlusion and complicated multi-objects dynamics, still needs to be investigated in the future. Future work also includes regulating the long-term memory size and improving template efficiency.

Acknowledgement

This work was supported by the National Science Foundation (Award #1636522) and the Monterey Bay Aquarium Research Institute.

References

[1] K. Katija, P. L. D. Roberts, J. Daniels, A. Lapides, K. Barnard, M. Risi, and B. Y. Ranaan, "Visual tracking of deepwater animals using machine learning-controlled robotic underwater vehicles", in *IEEE Winter Conference on Application of Computer Vision (WACV)*, pp.859-868, 2021.

[2] A. Azarbayejani and A. Pentland, "Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features," in Proceedings *13th International Conference on Pattern Recognition*, Vienna, Austria, pp. 627-632 vol.3, 1996.

[3] A. Islam, M. Asikuzzaman, M. O. Khyam, M. Noor-A-Rahim and M. R. Pickering, "Stereo Vision-Based 3D Positioning and Tracking," in *IEEE Access*, vol. 8, pp. 138771-138787, 2020.

[4] J. Rife and S. M. Rock, "Field experiments in the control of a Jellyfish tracking ROV," *OCEANS '02 MTS/IEEE*, pp. 2031-2038 vol.4, 2002.

[5] J. Rife and S. M. Rock, "A pilot-aid for rov based tracking of gelatinous animals in the midwater," in *MTS/IEEE Oceans 2001*, vol. 2, pp. 1137–1144 vol.2, 2001.

[6] J. Rife and S. M. Rock, "Visual tracking of jellyfish in situ," in Proceedings *International Conference on Image Processing*, pp. 289-292 vol.1, 2001.

[7] F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," in *CoRR*, vol. abs/1404.7584, 2014.

[8] M. Danelljan, G. H., F. S. Khan, and M. Felsberg, "Discriminativescale space tracking," in *CoRR*, vol. abs/1609.06141, 2016.

[9] A. Lukezic, T. Vojır, L. Cehovin, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *CoRR*,vol. abs/1611.08461, 2016.

[10] M. Danelljan, G. Häger, F. S. Khan and M. Felsberg, "Convolutional Features for Correlation Filter Based Visual Tracking," in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, pp. 621-629, 2015.

[11] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, J.-K.Kamarainen, L.ˇCehovin Zajc, M. Danelljan, A. Lukezic, O. Drbohlav,L. He, Y. Zhang, S. Yan, J. Yang, G. Fernandez, and et al., "The eighthvisual object tracking vot2020 challenge results," 2020.

[12] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H.Torr, "Fully-convolutional siamese networks for object tracking," in arXiv preprint arXiv:1606.09549, 2016.

[13] Z. Kalal, K. Mikolajczyk, and Jiri Matas, "Tracking-Learning-Detection", in *IEEE Pattern Analysis and Machine Intelligence*, 2012.

[14] H. Lee, S. Choi, and C.Kim, "A Memory Model based on the Siamese Network for Long-term Tracking", in *ECCV*, 2018.

[15] C. Ma, X. Yang, C. Zhang, and M. Yang, "Long-term correlation tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5388–5396, 2015.

[16] N. L. Baisa, D. Bhowmik, and A. Wallace, "Long-term correlation tracking using multi-layer hybrid features in sparse and dense environments," in *Journal of Visual Communication and Image Representation*, vol. 55, pp. 464 – 476, 2018.

[17] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao. "MUlti-Store Tracker (MUSTer): a Cognitive Psy-chology Inspired Approach to Object Tracking," in *CVPR*, 2015.

[18] A. Lukezic, L. C. Zajc, T. Voj ir, J. Matas, and M. Kristan, "A fully-correlational long-term tracker," ArXiv, vol. abs/1711.09594v2, 2019.

[19] C. Luo, B. Sun, K. Yang, Q. Deng, D. Jiang, "Part-based correlation filter tracking by exploiting the similarity and contribution of reliable parts," in *International Journal for Light and Electron Optics* 186 (2019) 165-176, 2019.

[20] A. Lukeźič, L. Č. Zajc, T. Vojíř, J. Matas and M. Kristan, "Performance Evaluation Methodology for Long-Term Single-Object Tracking," in *IEEE Transactions on Cybernetics*, 2020.

[21] L. Čehovin, M. Kristan and A. Leonardis, "Is my new tracker really better than yours?," in *IEEE Winter Conference on Applications of Computer Vision*, pp. 540-547, 2014.