Noname manuscript No.
(will be inserted by the editor)

Explaining Away Results In More Robust Visual Tracking

Bo Gao[1] · Michael W. Spratling[1]

Received: date / Accepted: date

Abstract Many current trackers utilise an appearance model to localise the target object in each frame. However, such approaches often fail when there are similar looking distractor objects in the surrounding background, meaning that target appearance alone is insufficient for robust tracking. In contrast, humans consider the distractor objects as additional visual cues, in order to infer the position of the target. Inspired by this observation, this paper proposes a novel tracking architecture in which not only is the appearance of the tracked object, but also the appearance of the distractors detected in previous frames, taken into consideration using a form of probabilistic inference known as explaining away. This mechanism increases the robustness of tracking by making it more likely that the target appearance model is matched to the true target, rather than similar-looking regions of the current frame. The proposed method can be combined with many existing trackers. Combining it with SiamFC, DaSiamRPN, Super_DiMP and ARSuper_DiMP all resulted in an increase in the tracking accuracy compared to that achieved by the underlying tracker alone. When combined with Super_DiMP and ARSuper_DiMP the resulting trackers produce performance that is competitive with the state-of-the-art on seven popular benchmarks.

Keywords Object tracking · Tracking-by-Detection trackers · Distractor-Submission · Explaining away

This research was funded by China Scholarship Council.

Bo Gao
ORCID: 0000-0003-3930-1815
E-mail: bo.gao@kcl.ac.uk

Michael W. Spratling
ORCID: 0000-0001-9531-2813
E-mail: michael.spratling@kcl.ac.uk

[1]Department of Informatics
King's College London
London, UK

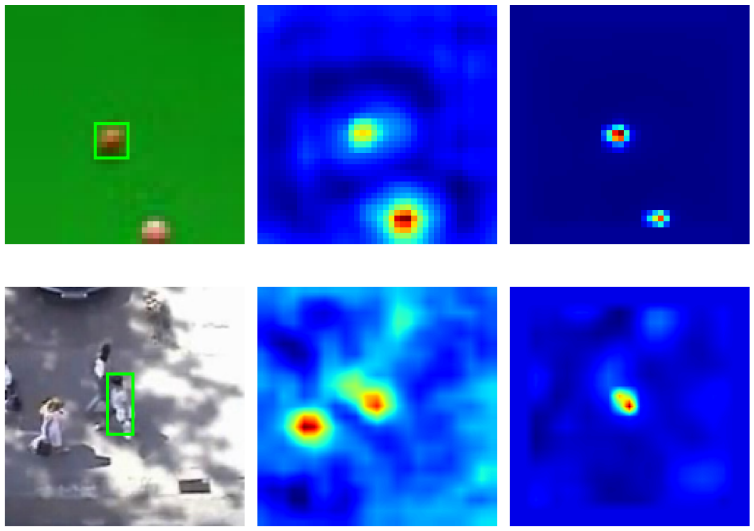
1 Introduction

Tracking is a fundamental task in computer vision with numerous applications in surveillance [41,47], self-driving vehicles [48,8], and UAV-based monitoring [43,59]. It is the task of locating the same moving object in each frame of a video sequence, given only the initial appearance of target object. Most modern trackers treat this as a classification problem. By learning an appearance model of the target from the initial frame, the trackers distinguish the target from background by cross-correlation operation and predict its location in the following frames. Although achieving impressive performance, these Tracking-by-Detection approaches can fail when the appearance model misidentifies a similar-looking object (a "distractor") as the target. Even a current state-of-the-art tracker, Super_DiMP[1], which fully exploits (through end-to-end offline training and online meta-learning) both target and background appearance information, is still fooled by distractors as shown in Fig. 1b. In contract, humans are able to take into account the appearance of other objects, in order to distinguish these potential distractors from the target object and successfully infer the position of a tracked object [19].

A leading theory of how such perceptual inference is performed in the brain is provided by predictive coding (PC) [55,49,53,9]. Specifically, PC suggests that the brain learns, from prior experience, an internal model of the world. This internal model encodes possible causes of sensory inputs and new sensory inputs are then represented in terms of these known causes. Determining which combination of the many possible causes best fits the current sensory data is achieved through an iterative process of minimising the error between the sensory data and the expected sensory inputs predicted by the causes [54]. This inference process performs explaining away [30,36,37]: possible causes (i.e. objects) compete to explain the sensory evidence (i.e. the current image), and if one cause explains part of the evidence (i.e. a part of the image), then support from this evidence for alternative explanations (i.e. other objects) is reduced, or explained away.

This paper proposes that explaining away, implemented using a version of PC called DIM [54,56], can be used to enable a tracker (like a human) to take into account the appearance of distractors when identifying the target in each frame of a video. Specifically, both the target and the distractors (identified in previous frames) are used as possible causes underlying the appearance for the next frame of the video. These causes compete to explain each part of the current frame, and when a distractor provides a better explanation for the appearance of some part of the image, this part of the image is explained away and will not be matched to the target. In this way, the target is less likely to be matched to incorrect, but similar-looking, regions of the image and is more likely to be matched to the correct location, increasing the robustness of tracking.

[1] Super_DiMP combines the bounding-box regressor of PrDiMP [15] with the standard DiMP classifier [3]. Code for this tracker is available at https://github.com/visionml/pytracking/



(a) Search region (b) Super_DiMP (c) Ours

Fig. 1 A comparison of our approach with a state-of-the-art tracker on two hard scenarios. Figure (a) shows the search region of the current frame and the green rectangle is the ground truth location of the tracked target object. Figure (b) and (c) are the score maps produced by Super_DiMP and by Super_DiMP when combined with the proposed method. Super_DiMP identified the wrong location as the most likely location for the target in both scenarios. Our approach correctly identifies the target location in both scenarios.

Our main contributions are summarised as follows:

- We propose a novel tracking architecture that detects distractors in every frame. These are represented as additional appearance models with the same size as the target appearance model. The predicted location of the target in the next frame takes into consideration not only the tracked object, but also the distractors. As a consequence matches between the target appearance model and the surrounding background are suppressed, and the identification of the target is more reliable (as illustrated in Fig. 1).
- The proposed method does not require any re-training of the underlying tracker and could easily integrate with most current trackers that use the Tracking-by-Detection architecture. We demonstrate this by integrating the proposed method with four existing trackers: SiamFC [2], DaSiamRPN [82], Super_DiMP [12], and ARSuper_DiMP [75]. In all cases the performance of the underlying tracker is improved by the addition of the proposed method. This indicates that the proposed method has good transferability and is potentially a general approach that could be used to improve the performance of most visual trackers.
- We demonstrate the effectiveness of our general approach by integrating it with the recent state-of-the-art trackers Super_DiMP [12] and ARSu-

per_DiMP [75]. The resulting trackers achieve results that are competitive with the state-of-the-art on seven benchmark datasets: OTB-100 [67], UAV123 [44], NFS [31], LaSOT [18], Trackingnet [45], GOT-10K [28] and VOT2020 [32].

2 Related Work

Contemporary approaches solve the tracking problem by learning the appearance of the target in the first frame. These approaches can be roughly divided into generative trackers [2,25,26,82,35,73,60,5,77,40,40,79,78,65,76,17] and discriminative trackers [46,13,12,27,7,14,39,3,15,4,23,63,11,75,64,81,68,40,79,80,71,72,69,70,38,42,16]. The former formulate object tracking as a cross correlation problem in deep feature space and take advantage of end-to-end learning by training a Y-shaped network containing two branches, one for the object template and the other for the search region. This approach is exemplified by Siamese network based trackers which have gained significant attention due to their promising performance and efficiency. However they typically employ a fixed target template and do not model background information, which consequently results in incorrect tracking when there is a similar looking object in the background or a significant change of the target appearance. Despite appearance updating strategies [25,60] that have been recently introduced into Siamese network based trackers, their performance is still below that of discriminative trackers.

The discriminative trackers are exemplified by discriminative correlation filter (DCF) based methods, which learn to distinguish the target from the background. Traditionally, these methods [27,14,39] have a fast training process in the Fourier domain using the diagonalizing transformation of circular convolutions to generate training samples. However, the online learning procedures are complicated and cannot be integrated with end-to-end learning architectures. To solve this problem, DiMP based trackers [3,15,11,4,75,64] employ a meta-learning formulation to predict the weights of the classification layer. This enables DiMP based trackers to achieve state-of-the-art results on many benchmarks. Despite discriminative trackers learning an appearance model using background information, the appearance model is still unable to deal with cases that contain highly similar looking distractors (as illustrated in Fig. 1).

Tracking failures caused by a similar-looking location being misidentified as the target, indicates that only using the appearance model to identify the tracked object is insufficient to achieve robust results for the popular Tracking-by-Detection based trackers. Some existing methods attempt to address this issue by taking more visual cues into consideration. For example, Gladh et al [23] use deep motion features extracted from optical flow images together with appearance features to generate the target model. Wang et al [63] predict the approximate location of the target by decoupling camera motion and object motion to create an adaptive search region.

Some existing methods attempt to address this issue by introducing attention mechanisms. For example, RAR [22] employed a hierarchical attention module to leverage both inter- and intra-frame attention at each convolutional layer which effectively highlighted informative representations and suppressed distractors. SiamGAT [24] employed a graph attention module to replace the cross-correlation operation, that is common in Siamese trackers, for part-to-part matching which effectively passed target information from the template to the search region. [64] proposed an appearance model generator using a transformer [61], the transformer-encoder promotes the previous appearance models via attention based feature reinforcement to acquire more compact target representations while the transformer-decoder generates the appearance model for the current frame. TransT [6] developed a Transformer-like fusion module to combine the template and search region features solely using attention instead of correlation. STMTrack [20] created a space-time memory network inspired by non-local self-attention [66] to fully use of historical information about the target to better adapt to appearance variations during tracking.

More closely related to our work are methods that explicitly take into account information about possible distractors. For example, DaSiamRPN [82] proposed a distractor-aware feature learning scheme to boost the discriminative power of the networks during off-line training, and also a novel distractor-aware module to suppress distractors during online tracking. Bhat et al [4] presented an end-to-end learning architecture, KYS, where the encoding of image regions is learned and propagated by appearance-based dense tracking between frames. The final prediction is then obtained by combining the explicit background representation with the appearance model output. Nocal-Siam [58] proposed a target-aware non-local attention module to jointly refine visual features of the target and search branches which suppressed distractors effectively.

Other distractor-suppression techniques have been proposed for specific tracking architectures, but would be difficult to incorporate in modern Tracking-by-Detection approaches. For example, [10] developed an on-line feature ranking mechanism to select the top-ranked appearance features for the trackers based on color histogram. TLD [29] proposed the Tracking-Learning-Detection architecture which implemented a P-N learning mechanism to exploit spatio-temporal relationships in the video. Siam R-CNN [62] proposed a Siamese re-detection architecture with a novel Tracklet Dynamic Programming Algorithm to simultaneously track all potential objects and select the best object in the current timestep based on the complete history of all target and distractor object tracklets. [74] proposed a novel hard negative mining method to suppress distractors for long-time tracking which enhanced the target identification ability of a verification network.

In contrast, we present a common distractor-suppression solution applicable to modern Tracking-by-Detection trackers. We design a novel architecture that constructs a joint appearance model for both the tracked and distractor objects. Each object in the appearance model then competes to explain each

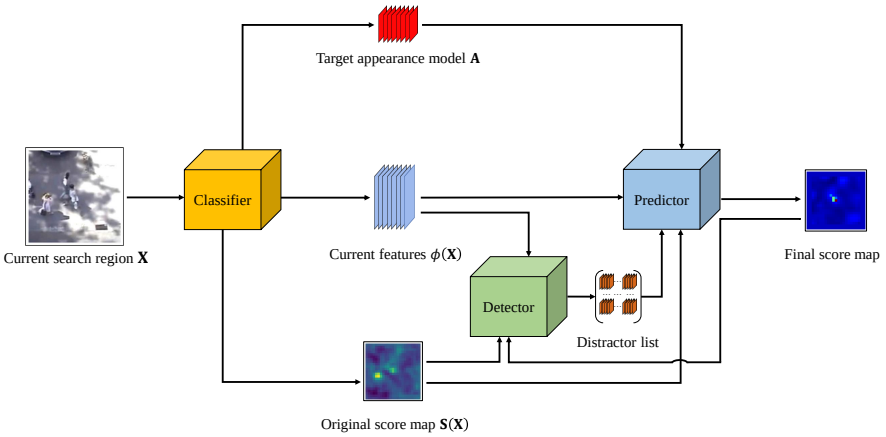


Fig. 2 An overview of the proposed tracking architecture. Distractor objects are located in every frame by identifying non-target peaks in the score map generated by the tracker. Distractor appearance models are obtained by cropping areas from the current image features $\phi(\mathbf{X})$, that are the same size as the target appearance model but centered at the distractor positions. The distractor list contains the distractor appearance models detected in the previous n frames. These models together with the target model are united as a joint appearance model for our predictor to compute for the best matching location.

part of the next video frame. This leads to the score map for the target being suppressed at locations where the appearance of a distractor is a better match to the image, and consequently results in more robust predictions about the true location of the target.

3 Proposed Method

Fig. 2 shows the architecture of the proposed method. A visual tracker is used to generate an initial prediction which is then used by the proposed detector module (see Section 3.1 for details) to locate distractor objects. Once the positions of distractors are determined, the corresponding distractor appearance models are obtained by cropping regions from the current image features. Lastly, the proposed predictor module (see Section 3.2 for details) takes the distractor appearance models (detected in previous frames) and the target appearance model into consideration at same time. These models compete to explain every pixel of the image, which results in the suppression of distractors in the final score map, which describes the similarity between the target and each location in the image.

3.1 Distractor detection

During tracking, both generative and discriminative trackers (see Section 2) predict a scalar confidence score map $\mathbf{S}(\mathbf{X}) \in \mathbb{R}$ given an input image $\mathbf{X}$, such that:

$$\mathbf{S}(\mathbf{X}) = \mathbf{A} \star \phi(\mathbf{X}) \tag{1}$$

Here, $\phi(\mathbf{X})$ are the features extracted from the search region of the image, commonly by a CNN. $\mathbf{A}$ is the target appearance model, for Siamese trackers it represents the features of the template $\mathbf{Z}$, i.e. $\phi(\mathbf{Z})$. For DCF trackers it is the convolution kernel which is trained online. $\star$ represents the cross-correlation operation.

The score map measures the similarity between an appearance model and the deep features extracted from the current video frame. The tracker estimates the target object's location by finding the location of the maximum in the score map. If the appearance of the target is distinctive there will only be one peak in the score map. However, if there are similar looking distractors in the search region, the score map will have multiple peaks. Hence, distractors can be identified by finding the locations of peaks excluding the one that represents the target. To be specific, a peak is defined as a local maximum (within a 3-by-3 neighbourhood) that has a value over a global threshold which is set to 0.7 times the max value of the score map. The peak corresponding to the target is determined by finding the location of the maximum value in the final score map produced by the proposed predictor.

Finally, distractor appearance models are obtained by cropping areas from $\phi(\mathbf{X})$ that are the same size as the target appearance model, and are centered at the distractor positions. A list of distractors is updated every frame and contains the distractor appearance models detected in the last n (default value is 5) frames. If there is no distractor in a frame, no additional distractor appearance models are stored in the list.

3.2 Appearance model competition

The predictor takes the target appearance model and the distractor appearance models extracted from the preceding n frames. These appearance models compete to match to the features extracted from the search region by the tracker. The competition is achieved by the DIM algorithm which implements explaining away and which is the current state-of-the-art method for image patch matching in both color [56] and deep feature [21] space. A detailed description of the DIM algorithm can be found in [56], but an introduction is provided below for the convenience of the reader.

The DIM can be thought of as a function, with two input arguments and one output. In the current application this function operates as follows:

$$\mathbf{S}_j(\mathbf{X}) = \mathrm{DIM}(Pre(\mathbf{A}_j), Pre(\phi(\mathbf{X}))_{\epsilon_2,\iota} \tag{2}$$

Where $\mathbf{A}_j$ is a joint appearance model consisting of a stack of the target appearance model and the distractor appearance models detected in last n frames, $\phi(\mathbf{X})$ are the features extracted from the search region of the image, $\mathbf{X}$. The output of this function, $\mathbf{S}_j(\mathbf{X})$, is a stack of score maps, each channel, j, is the individual score map for the corresponding appearance model in $\mathbf{A}_j$. Pre stands for pre-processing and will be described below.

To simplify the notation, we will represent the two inputs to DIM as $\mathbf{w}$ and $\mathbf{I}$ (i.e. $\mathbf{w}_j = Pre(\mathbf{A}_j)$ and $\mathbf{I} = Pre(\phi(\mathbf{X}))$. Internally, the DIM function performs ι iterations for the following three equations:

$$\mathbf{R}_i = \sum_{j=1}^{p} (\mathbf{v}_{ji} \star \mathbf{S}_j) \tag{3}$$

$$\mathbf{E}_i = \mathbf{I}_i \oslash [\mathbf{R}_i]_{\epsilon_2} \tag{4}$$

$$\mathbf{S}_j \leftarrow [\mathbf{S}_j]_{\epsilon_1} \odot \sum_{i=1}^{k} (\mathbf{w}_{ji} * \mathbf{E}_i) \tag{5}$$

Where i is an index over the number of channels in the input $\mathbf{I}$; j is an index over the number of different appearance models; $\mathbf{R}_i$ is a 2-dimensional array representing a reconstruction of $\mathbf{I}_i$; $\mathbf{E}_i$ is a 2-dimensional array representing the discrepancy (or residual error) between $\mathbf{I}_i$ and $\mathbf{R}_i$; $\mathbf{S}_j$ is the individual score map for the corresponding appearance model in $\mathbf{A}_j$; $\mathbf{w}_{ji}$ is a 2-dimensional array representing channel i of the corresponding appearance model after pre-processing (i.e. $Pre(\mathbf{A}_j)_i$) the values in each $\mathbf{w}_j$ were normalised to sum to one; $\mathbf{v}_{ji}$ is a 2-dimensional array also representing appearance model values (the values of $\mathbf{v}_j$ were made equal to the corresponding values of $\mathbf{w}_j$ except they were normalised to have a maximum value of one); $[\cdot]_\epsilon = max(\cdot, \epsilon)$; $\oslash$ and $\odot$ indicate element-wise division and multiplication respectively; $\star$ and $*$ represent cross-correlation and convolution operations respectively.

DIM attempts to find a sparse set of elementary components, $\mathbf{v}$, that when combined together reconstruct $\mathbf{I}$ with minimum error [52]. For the current application, the elementary components are the target appearance model and the distractor appearance models in the distractor list. These appearance models can be thought of as a "dictionary" or "codebook" that can be used to reconstruct many different images. The activation dynamics, described by Eqs. 3, 4 and 5, perform gradient descent on the residual error in order to find values of $\mathbf{S}$ that accurately reconstruct $\mathbf{I}$ [1,51,57]. Specifically, the equations operate to find values for $\mathbf{S}$ that minimise the Kullback-Leibler (KL) divergence between $\mathbf{I}$ and its reconstruction $\mathbf{R}$ [50,57]. The activation dynamics thus result in the DIM algorithm selecting a subset of dictionary elements that best explain $\mathbf{I}$. The strength of an element in $\mathbf{S}$ reflects the strength with which the corresponding dictionary entry (i.e. appearance model) is required to be present in order to accurately reconstruct $\mathbf{I}$ at that location [56]. Hence when a distractor appearance model provides a high similarity to the appearance of some part

of the image, this part of the image is explained away and will not be matched to the target. In this way, the target appearance model is more likely to be matched to the correct location, increasing the robustness of tracking.

Because DIM minimises the KL divergence between $\mathbf{I}$ and its reconstruction $\mathbf{R}$ created by the additive combination of elementary image components $\mathbf{v}$, both inputs to the DIM function must be non-negative [56]. However $\mathbf{A}_j$ and $\phi(\mathbf{X})$ are activation values extracted from a CNN which can contain negative values. Thus the pre-processing takes the positive and rectified negative values of $\mathbf{A}_j$ and $\phi(\mathbf{X})$ and separates them into two parts which are used as separate channels by the DIM algorithm.

ϵ_1 is a parameters which was given to the values $\frac{\epsilon_2}{\max(\sum_j^p v_{ji})}$. ϵ_2 is a scalar parameter used by the DIM algorithm. It determines the magnitude required elements of $\mathbf{I}$ to have a strong effect on the competition. Hence, if a value of $\mathbf{I}$ is smaller than ϵ_2 it is effectively ignored. When DIM is applied to colour images, those images have pixel intensities that typically range from 0 to 1, so the maximum value of $\mathbf{I}$ is approximately 1 for every image, and it is possible to use a fixed value of ϵ_2. However the maximum value of $Pre(\phi(\mathbf{X}))$, which is used as $\mathbf{I}$, can vary as it is produced by applying a CNN to different videos. To deal with this variation the appropriate value for ϵ_2 for any one video is chosen from a set of ten possible values: values ranging from 1×10^{-3} to 9×10^{-3} in steps of 8×10^{-4}. When DIM is applied for the first time to a video it is run ten times with each candidate ϵ_2 value. The magnitude of the highest peaks in the resulting ten score maps for the target object are compared, and the value of ϵ_2 corresponding to the highest peak is used for all subsequent frames of this video. The number of iterations, ι, performed by the DIM algorithm was set to 15.

3.3 Implementation Details

DIM requires the appearance model to have dimensions that are odd numbers, otherwise the reconstruction of $\mathbf{I}$ does not align with the actual $\mathbf{I}$. Therefore, if the size of the target appearance model employed by a visual tracker is even, the target appearance model is padded by one row on the right and one column on the bottom with zeros and the new size of target appearance model is used to generate distractor appearance models, as described in Section 3.1.

If no distractor appearance model has been detected in the preceding five frames, i.e. if $\mathbf{A}_j$ only contains the target appearance model, then DIM will output a similar result to that produced by Eq. 1. In such circumstances, when the distractor list is empty, DIM is not employed and the score map generated by the original tracker is used as the final score map. This helps to improve the computational efficiency of the proposed method. The frequency of DIM used in the trackers reported in this paper can be found in Section 4.5.

Tracking sometimes fails, for example, when the target is occluded or is out of frame. In this case, the tracker may confuse a distractor for the target. Sub-

Algorithm 1 Proposed Tracker.

Input: The first frame and the ground-truth bounding box of the target in first frame, the current frame im_t;
Output: The predicted target bounding box in every frame;
Initialize underlying tracker;
for $t = 2$ to k (the total number of frames) **do**
  Generate the score map $\mathbf{S}(\mathbf{X})$[2]using the classifier in the underlying tracker;
  Find the number, m, of peaks above threshold g in $\mathbf{S}(\mathbf{X})$ and their locations, $locs_t$;
  if $m > 1$ and distractor list is not empty **then**:
    Update $\mathbf{S}(\mathbf{X})$ using DIM with ι iterations;
  end if
  Find location of the highest peak, $hloc_t$, in $\mathbf{S}(\mathbf{X})$;
  if $\|hloc_t, hloc_{t-1}\|_2 > d$ **then**
    Clear the distractor list and do not update it for next r frames;
  else
    Determine the locations of distractors by excluding $hloc_t$ in $locs_t$;
    Crop distractor appearance models from $\phi(\mathbf{X})$;
    Update the distractor list using the distractor appearance models;
  end if
  Estimate the predicted bounding box using $\mathbf{S}(\mathbf{X})$ using the estimator in the underlying tracker;
end for

sequently, when the tracked object reappears the proposed detection module will incorrectly regard the reappeared target object as a distractor due to the incorrect matching in the former frame. If this happens the appearance of the target object will be included in the list of distractor appearance models and DIM will suppress the score map at the location of the true target. To avoid this phenomenon, we rely on the assumption that the position of target object between two adjacent frames doesn't change significantly, while there will be a jump in the predicted position of the target when a distractor is confused for the target. Specifically, the Euclidean distance between the locations of the highest peaks of the final score map in this frame and the one in the former frame is calculated. If this distance exceeds a threshold d (a value of 3 was used and the distance was calculated before upsampling the score map) the distractor list is cleared and the proposed predictor does not run for r frames (a value of 5 was used). Hence, for those r frames the final score map will be the one generated by the underlying tracker.

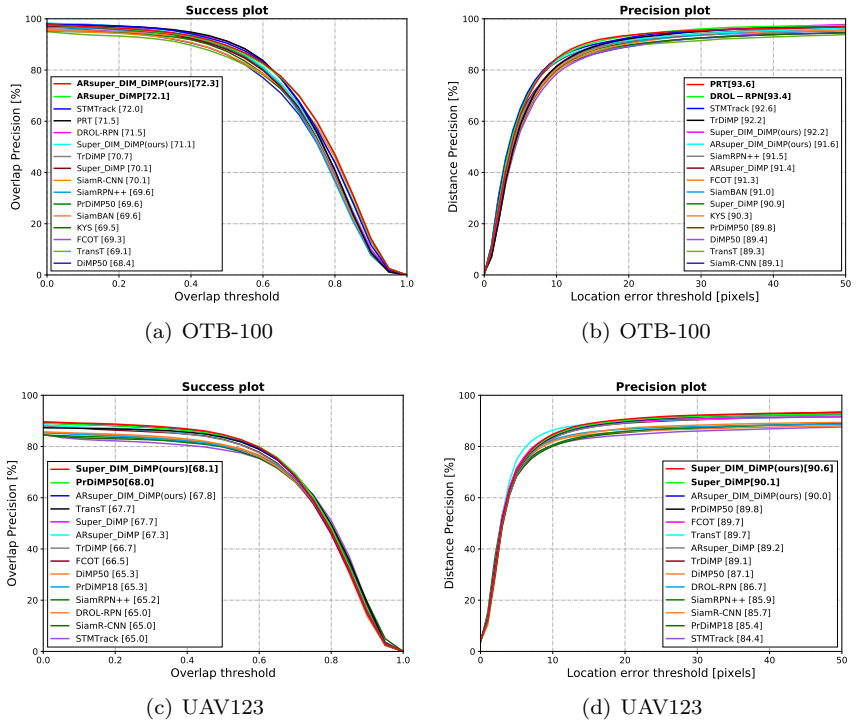
The complete proposed architecture is summarised by Algorithm 1.

4 Experiments

4.1 State-of-the-art Comparison

We evaluate our proposed tracking architecture using Super_DiMP [12] and ARSuper_DiMP [75] on seven tracking benchmarks: OTB-100 [67], UAV123

[2] $\mathbf{X}$ is the current search region obtained by cropping a square area in im_t to search for the target around the location at which it appeared in im_{t-1}.



(a) OTB-100 (b) OTB-100

(c) UAV123 (d) UAV123

Fig. 3 Success and Precision measured using the OTB-100 and UAV123 datasets.

[44], NFS [31], LaSOT [18], Trackingnet [45], GOT-10k [28] and VOT2020 [32]. Due to the stochastic nature of DCF trackers, the results reported for DiMP-based trackers [3, 15, 4, 75, 64] are an average over multiple runs. For OTB-100, NFS, UAV123, and LaSOT, the results were averaged over five runs. As the results of Trackingnet are obtained using an online evaluation server with limited submissions for an account, only a single run was used. GOT-10k results are also evaluated with an online server and the official documentation suggests using three runs for all trackers, hence three runs were used. The official VOT evaluation toolkit runs a tracker twenty times as default to produce statistically significant results, hence, twenty runs were used for VOT2020. For a fair comparison, we follow the same approach to test our trackers, termed Super_DIM_DiMP and ARSuper_DIM_DiMP, and the original trackers. As ARSuper_DiMP uses an Alpha-Refine module to improve the accuracy of the bounding boxes predicted by Super_DiMP, the hyper-parameters used by our method were tuned using Super_DiMP and re-used for ARSuper_DiMP. Super_DIM_DiMP runs at around 23 FPS on a single Nvidia Tesla V100 GPU. In comparison, Super_DiMP runs at approximately 27 FPS. ARSuper_DIM_DiMP and ARSuper_DiMP run at around 20 and 16 FPS respectively. Our code will be released upon publication.

Arch.	Tracker	Succ.	Prec.
Tracking-By-Detection	ATOM [13]	58.0	70.0
	FCOT [11]	63.2	76.1
	DiMP50 [3]	61.5	74.1
	PrDiMP50 [15]	63.5	75.9
	KYS [4]	63.3	76.1
	SiamBAN [7]	59.5	70.0
	TrDiMP [64]	65.8	79.1
	Super_DiMP [12]	64.7	78.1
	ARSuper_DiMP [75]	66.3	80.5
	Super_DIM_DiMP (ours)	65.5	79.5
	ARSuper_DIM_DiMP (ours)	67.5	82.3
Others	STMTrack [20]	-	-
	TransT [6]	65.7	-
	Siam R-CNN [62]	63.9	-

Table 1 Comparison with state-of-the-art trackers on the NFS dataset. The best two results are highlighted by red and green.

OTB-100 [67]: This dataset has been used extensively to evaluate visual trackers. Our methods are compared with numerous state-of-the-art trackers in Fig. 3a and 3b, including STMTrack [20], PRT [40], DROL-RPN [79], ARSuper_DiMP [75], TrDiMP[3] [64], Super_DiMP [12], SiamR-CNN [62], SiamRPN++ [35], PrDiMP50 [15], SiamBAN [7], KYS [4], FCOT [11], TransT [6] and DiMP50 [3]. Despite performance becoming saturated over recent years the proposed tracker Super_DIM _DiMP still outperforms the baseline, Super_DiMP, by 1% in terms of AUC (success score) and 1.3% in terms of precision. Similarly, ARSuper_DIM_DiMP outperforms ARSuper_DiMP in both scores and achieves the best AUC score with 72.3%.

UAV123 [44]: This is a large dataset captured from low-altitude UAVs. It contains over 110K frames and 123 videos. It is quite changing due to small tracked objects and fast motion. PrDiMP50 [15], TransT [6], Super_DiMP [12], ARSuper_DiMP [75], TrDiMP [64], FCOT [11], DiMP50 [3], PrDiMP18 [15], SiamRPN++ [35], DROL-RPN [79], SiamR-CNN [62], STMTrack [20], DiMP18 [3] are compared. It can be seen from the results shown in Fig. 3c and 3d that Super_DIM_DiMP outperforms the previous best approaches with an AUC of 68.1% and precision of 90.6%.

NFS [31]: This dataset contains 100 videos captured using a high frame rate (240 FPS) camera. We evaluate our tracker on the 30 FPS version of this dataset in which videos have an average length of 479 frames. As shown in Table 1, ARSuper_DIM_DiMP outperforms the previous best approaches with an AUC of 67.5% and precision of 82.3%.

LaSOT [18]: The large-scale LaSOT dataset contains 280 videos in its test set. The video sequences, which have an average length of 2500 frames, are longer than those in other datasets, testing not only the accuracy of the tracker

[3] Using the raw results provided by the authors, we were unable to reproduce the scores reported for TrDiMP in [64] for the OTB-100, UAV123 and NFS datasets. Our different results are shown in Fig. 3 and Table 1.

Arch.	Tracker	Succ.	P_{norm}
Tracking-By-Detection	SiamFC++ [73]	55.7	58.9
	ATOM [13]	51.5	57.6
	Nocal-Siam [58]	53.3	-
	FCOT [11]	56.9	67.8
	DiMP50 [3]	56.9	65.0
	PrDiMP50 [15]	59.8	68.8
	SiamBAN [7]	51.4	59.8
	TrDiMP [64]	63.9	-
	Super_DiMP [12]	63.0	71.9
	ARSuper_DiMP [75]	65.3	73.6
	Super_DIM_DiMP (ours)	63.4	72.5
	ARSuper_DIM_DiMP (ours)	65.5	73.4
Others	STMTrack [20]	60.6	69.3
	TransT [6]	64.9	73.8
	Siam R-CNN [62]	64.8	72.2

Table 2 Comparison with state-of-the-art trackers on the LaSOT dataset. The best two results are highlighted by red and green.

but also its robustness. As shown in Table 2, ARSuper_DIM_DiMP achieves the best AUC score with 65.5% and Super_DIM_DiMP outperforms Super_DiMP with relative gains of 0.4% in success score and 0.6% in normalized precision.

Trackingnet [45]: This dataset provides over 30K videos sampled from YouTube. We report results on its test set, consisting of 511 videos with an average of 441 frames per sequence. As shown in Table 3, our approaches achieve similar results to the baseline tackers Super_DiMP and ARSuper_DiMP.

GOT-10K [28]: This is a recent large-scale dataset consisting of 10k video sequences. With this dataset trackers are evaluated on 180 videos with 84 object classes and 32 motions that cover a wide range of common moving objects in the wild. The results, in term of average overlap (AO) and success rates ($SR_{0.50}$ and $SR_{0.75}$[4]) are shown in Table 4. Among Tracking-By-Detection methods, ARSuper_DIM_DiMP achieves the performance. Meanwhile, Super_DIM_DiMP significantly outperforms Super_DiMP with a relative improvement of 2.5% in AO.

VOT2020 [32]: The VOT challenge [33,34,32], held yearly, provides a precisely defined and repeatable way of comparing short-term trackers. VOT2020 contains 60 videos with binary segmentation masks as the ground-truth and uses a new evaluation protocol which separates the sequences into short pieces to keep the computational complexity of the evaluation at a moderate level. The results, in term of expected average overlap (EAO), accuracy (A) and robustness (R) are shown in Table 4. ARSuper_DIM_DiMP outperforms the baseline ARSuper_DiMP by 1.2% in term of robustness.

Of all the Tracking-By-Detection trackers tested, ARSuper_DIM_DiMP was the best performing on five datasets (OTB-100, NFS, LaSOT, Trackingnet, and GOT-10k). No other Tracking-By-Detection method was best performing

[4] The percentage of successfully tracked frames where overlap rates are above the given threshold.

Arch.	Tracker	Prec.	Pnorm	Succ.
Tracking-By-Detection	SiamFC++ [73]	70.5	80.0	75.4
	ATOM [13]	64.8	77.1	70.3
	SiamRPN++ [35]	69.4	80.0	73.3
	FCOT [11]	72.3	82.8	75.1
	DiMP50 [3]	68.7	80.1	74.0
	PrDiMP50 [15]	70.4	81.6	75.8
	KYS [4]	68.8	80.0	74.0
	TrDiMP [64]	73.1	83.3	78.4
	Super_DiMP [12]	73.3	83.4	78.1
	ARSuper_DiMP [75]	78.3	85.6	80.5
	Super_DIM_DiMP (ours)	73.5	83.7	78.2
	ARSuper_DIM_DiMP (ours)	78.5	85.8	80.6
Others	STMTrack [20]	76.7	85.1	80.3
	TransT [6]	80.3	86.7	81.4
	Siam R-CNN [62]	80.0	85.4	81.2

Table 3 Comparison with state-of-the-art trackers on the Trackingnet dataset. The best two results are highlighted by red and green.

Arch.	Tracker	AO	$SR_{0.5}$	$SR_{0.75}$
Tracking-By-Detection	Nocal-Siam [58]	60.1	68.8	-
	FCOT [11]	64.0	76.3	51.7
	DiMP50 [3]	61.1	71.7	49.2
	PrDiMP50 [15]	63.4	73.8	54.3
	KYS [4]	63.6	75.1	51.5
	TrDiMP [64]	68.8	80.5	59.7
	Super_DiMP [12]	67.5	78.8	59.5
	ARSuper_DiMP [75]	70.1	80.0	64.2
	Super_DIM_DiMP (ours)	69.2	80.8	60.6
	ARSuper_DIM_DiMP (ours)	70.9	80.9	64.7
Others	STMTrack [20]	64.2	73.7	57.5
	TransT [6]	72.3	82.4	68.2
	Siam R-CNN [62]	64.9	72.8	59.7

Table 4 Comparison with state-of-the-art trackers on the GOT-10K dataset. The best two results are highlighted by red and green.

Arch.	Tracker	EAO	A	R
Tracking-By-Detection	ATOM [13]	27.1	46.2	73.4
	DiMP50 [3]	27.4	45.7	74.0
	OceanPlus [76]	49.1	70.0	74.2
	PRT [40]	53.0	70.0	86.9
	Ocean [78]	43.0	69.3	75.4
	Siammask [65]	32.1	62.4	64.8
	D3S [38]	43.9	69.9	76.9
	Super_DiMP [12]	30.5	47.7	78.6
	ARSuper_DiMP [75]	48.2	75.4	77.7
	Super_DIM_DiMP (ours)	30.6	47.8	78.8
	ARSuper_DIM_DiMP (ours)	48.5	72.7	78.9
Others	TransT [6]	48.5	-	-
	Siam R-CNN [62]	35.5	69.9	58.6

Table 5 Comparison with state-of-the-art trackers on the VOT2020 dataset. The best two results are highlighted by red and green.

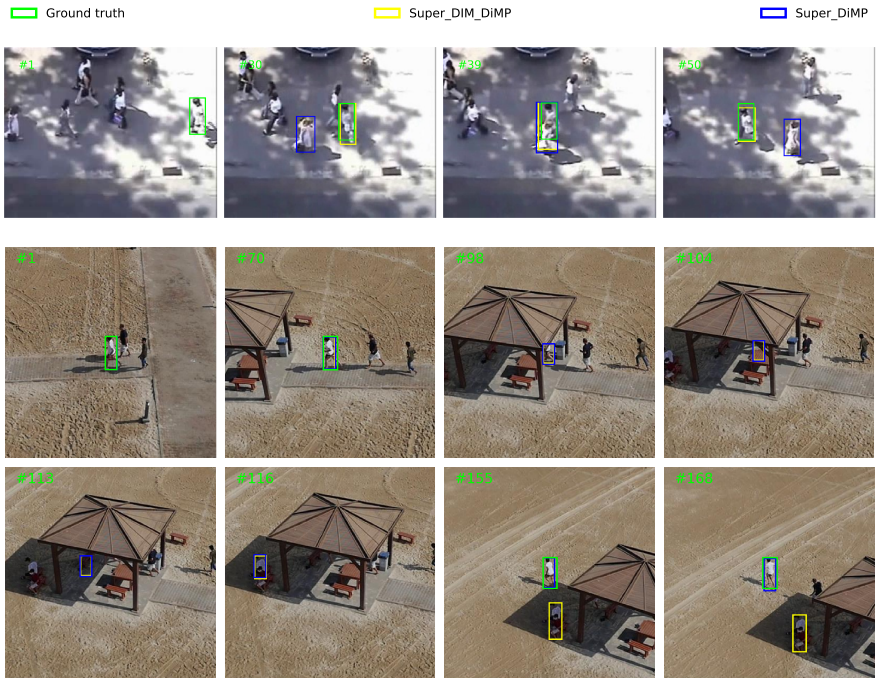


Fig. 4 Qualitative comparison of Super_DiMP and Super_DIM_DiMP on hard scenarios. First row shows frames from video *Crowds* from OTB-100, second and third rows are frames from video *group2_1* from UAV123. Note that the annotations provided with the UAV123 data, define the bounding box coordinates as NaNs when the target is out-of-frame or fully occluded, thus the ground truth bounding box is not shown in frame 98, 104, 113, and 116 of the video *group2_1*. Note that cropped regions of each video are shown to improve the visibility of the bounding boxes.

on more than one dataset, and for one of those (UAV123) the best performing tracker was our other method, Super_DIM_DiMP. The results also show DIM to be more effective than other recent distractor-suppression methods such as KYS [4] and Nocal-Siam [58].

4.2 Qualitative evaluations

The effects of explaining away are illustrated on the first row of Fig. 4. In this example, Super_DiMP incorrectly starts to track a similar looking distractor in frame 30. In contrast, this distractor was detected in previous frames and the proposed tracker is able to use this information to infer the true location of the target. However, the proposed tracker can fail when the tracking object is fully covered for a long time. In the example on the second and third rows, the target is fully occluded by a pavilion from frame 98 to 141 (four frames are selected in Fig. 4). Super_DiMP checks the maximum of $\mathbf{S}(\mathbf{X})$ every frame,

Parameter	Standard value	AUC when value changed by: ÷5	÷2	×2(1.15)	×5(1.3)
g	0.7	68.6	70.3	71.2	70.3
n	5	70.3	70.6	70.9	70.8
ι	15	70.5	70.7	70.2	70.4
d	3	70.2	70.9	69.8	68.0
r	5	70.2	70.7	70.8	70.5

Table 6 Evaluation of the sensitivity of the proposed architecture to its parameter values on OTB-100. Note the value of g is no more than 1, thus the factors in brackets are used instead. n, ι, d and r need to be integers, so are rounded up to the nearest integer value. Using Super_DiMP alone the AUC is equal to 70.1. Using the proposed method with the standard parameter values the AUC is equal to 71.1.

if the value is below a threshold that is interpreted as a tracking failure, the tracker does not update the location of the target, and outputs the same predicted bounding box as the last frame. Hence the predicted bounding box of the two trackers on frame 98, 104 and 113 are highly overlapped. From frame 116 the proposed tracker regards a distractor as the target even when the target reappears. This is because the proposed detection module incorrectly regards the reappeared target object as a distractor, and the appearance of the target object is included in the list of distractor appearance models. Hence, DIM suppresses the score map at the location of the true target. We have developed a reset mechanism to avoid this phenomenon (see Section 3.3 for details), however this mechanism is not activated in this particular scenario as the proposed tracker regards the distractor as the target for a long time. However, this type of failure case (which requires the tracking target to be fully occluded for a long time, and the surrounding background to contain a highly similar distractor) is rare, and hence, has little detrimental effect on performance overall.

4.3 Parameter Sensitivity

The proposed method employs a number of hyper-parameters:

- The global threshold, g, applied to the score map to locate peaks caused by distractors, see Section 3.1;
- The number of previous frames, n, used to detect distractors, see Section 3.1;
- The number of iterations, ι, performed by the DIM algorithm, see Section 3.2;
- The distance threshold, d, used to identify situations where the target has been lost, see Section 3.3;
- The reset period, r, during which DIM is not used following the distance threshold being exceeded, see Section 3.3;

The influence of these hyper-parameters on the performance of Super_DIM_DiMP was evaluated by varying the value of one parameter while keeping the other

parameters fixed at their default values. This experiment was conducted using the OTB-100 dataset, and the results are shown in Table 6.

It can be seen that when the value of g was increased by a factor of 1.15 from its default value, the algorithm still produced state-of-the art performance. However, increasing this parameter further had a detrimental effect on performance, which is not surprising as very few distractors will be identified if g is too large. Decreasing g also reduced performance, and an extreme reduction in g could lead to worse performance than the underlying tracker alone. This is likely to be due to the DIM algorithm needing more iterations to perform explaining away when there are many distractor appearance models. In addition, small amplitude peaks in the score map close to the target will result in parts of the target being included in the distractor appearance models.

The algorithm was tolerant to large changes in n, ι and r. However, only detecting distractors in one preceding frame ($n \div 5$) meant that there were few distractor appearance models, and hence, only a minor performance gain compared to the underlying tracker alone. Performance deteriorated when a large number of iterations was performed, this can be explained by the similarity values becoming sparser as the number of iterations increase [56]. Using a vary small r resulted in an AUC only marginally above that of the base tracker which indicates that one frame was insufficient time for the tracker to re-locate the target.

The results were particularly sensitive to the value of d. When d was decreased by a factor of 5 from its default value, the proposed method produced similar results to the underlying tracker alone. This is because the small distance threshold excluded DIM from being used most of the time, due to small displacements of the target from one frame to the next. In contrast, increasing the value of d meant the situations where the target was lost were not identified, and as explained in Section 3.3, this can result in target object being included among the distractor appearance models, and the score map being suppressed at the true location of the target in subsequent frames.

4.4 Transferability

The experiments above are based on the discriminative trackers, Super_DiMP [12] and ARSuper_DiMP [75]. To test the transferability of our method, we also combined it with two generative trackers: SiamFC [2] and DaSiamRPN [82]. The global threshold g was set to 0.5 for these two trackers and other hyper-parameters were kept at the standard values. Optimising the parameters carefully for each tracker may result in better performance. The resulting methods were evaluated on the OTB-100 and UAV123 dataset, and the results are shown in Table 7. We evaluated them on a single Nvidia Tesla V100 GPU. The speed of SiamFC and DaSiamRPN are around 210 FPS and 103 FPS respectively. When integrating DIM, their speeds are 175 FPS and 82 FPS respectively. The results show our method can also improve the tracking per-

Tracker		OTB-100		UAV123	
		Succ.	Prec.	Succ.	Prec.
SiamFC	Original	58.1	79.1	50.2	70.4
	Proposed	58.3	79.4	50.5	70.8
	Increment	0.2	0.3	0.3	0.4
DaSiamRPN	Original	63.7	84.5	60.6	77.4
	Proposed	64.5	85.6	61.0	78.3
	Increment	0.8	1.1	0.4	0.9

Table 7 Evaluation of proposed architecture with generative trackers. Note that the authors of DaSiamRPN [82] report an online update module for the target template but they did not release this as part of the official implementation. Thus the results reported here are lower than those in [82].

formance of both these trackers, which indicates our method has the potential to be a general approach that could improve the performance for most visual trackers.

4.5 The frequency of DIM used in these trackers

We tested this using OTB-100. This dataset contains 59040 frames. DIM was used on 1817, 1723, 15757, and 13244 frames when integrated with Super_DiMP, ARSuper_DiMP, SiamFC and DaSiamRPN respectively. DIM is used less frequently in Super_DiMP and ARSuper_DiMP as these trackers are much more robust than the other two trackers and also because of the use of a higher global threshold g (0.7 of these trackers and 0.5 of other two trackers). Because Super_DiMP and ARSuper_DiMP fully exploit both target and background appearance information during tracking, they are only fooled on rare occasions when a distractor is highly similar to the tracked object. DIM works on these rare occurrences to provide a useful boost in performance.

5 Conclusions

We propose a novel tracking architecture that can detect distractors in each frame of a video. The distractor appearance models compete with the target appearance model to explain each part of a subsequent frame of the video. Parts of the image that look similar to the target, and might have been misidentified as the target, are explained away by the distractor appearance models. This leads to suppression of the distractors in the score map, and hence, to more robust tracking of the target. It is general-purpose and has the potential to improve the performance of many exiting tracking algorithms, and when combined with state-of-the-art discriminative trackers are shown to improve tracking results even further.

Acknowledgements The authors acknowledge use of the research computing facility at King's College London, Rosalind (https://rosalind.kcl.ac.uk), and the Joint Academic Data science Endeavour (JADE) facility.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Achler, T.: Symbolic neural networks for cognitive capacities. Biologically Inspired Cognitive Architectures **9**, 71–81 (2014) 8
2. 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 (ECCV), pp. 850–865. Springer (2016) 3, 4, 17
3. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 6182–6191 (2019) 2, 4, 11, 12, 13, 14
4. Bhat, G., Danelljan, M., Van Gool, L., Timofte, R.: Know your surroundings: Exploiting scene information for object tracking. arXiv:2003.11014 (2020) 4, 5, 11, 12, 14, 15
5. Bo, L., Junjie, Y., Wei, W., Zheng, Z., Xiaolin, H.: High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8971–8980 (2018) 4
6. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8126–8135 (2021) 5, 12, 13, 14
7. Chen, Z., Zhong, B., Li, G., Zhang, S., Ji, R.: Siamese box adaptive network for visual tracking. arXiv:2003.06761 (2020) 4, 12, 13
8. Cho, H., Seo, Y.W., Kumar, B.V., Rajkumar, R.R.: A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 1836–1843. IEEE (2014) 2
9. Clark, A.: Whatever next? predictive brains, situated agents, and the future of cognitive science. Behavioral and Brain sciences **36**(3), 181–204 (2013) 2
10. Collins, R.T., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. IEEE transactions on pattern analysis and machine intelligence **27**(10), 1631–1643 (2005) 5
11. Cui, Y., Jiang, C., Wang, L., Wu, G.: Fully convolutional online tracking. arXiv:2004.07109 (2020) 4, 12, 13, 14
12. Danelljan, M., Bhat, G.: Pytracking: Visual tracking library based on pytorch (2019). https://github.com/visionml/pytracking/,accessed: 6/01/2020 3, 4, 10, 12, 13, 14, 17
13. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ATOM: Accurate tracking by overlap maximization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4660–4669 (2019) 4, 12, 13, 14
14. Danelljan, M., Bhat, G., Shahbaz Khan, F., Felsberg, M.: ECO: Efficient convolution operators for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6638–6646 (2017) 4
15. Danelljan, M., Gool, L.V., Timofte, R.: Probabilistic regression for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7183–7192 (2020) 2, 4, 11, 12, 13, 14
16. Devi, R.B., Chanu, Y.J., Singh, K.M.: Discriminative object tracking with subspace representation. The Visual Computer **37**, 1207–1219 (2021) 4
17. Fan, C., Zhang, R., Ming, Y.: Mp-ln: motion state prediction and localization network for visual object tracking. The Visual Computer pp. 1–16 (2021) 4

18. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: Lasot: A high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5374–5383 (2019) 4, 11, 12
19. Feria, C.S.: The effects of distractors in multiple object tracking are modulated by the similarity of distractor and target features. Perception **41**(3), 287–304 (2012) 2
20. Fu, Z., Liu, Q., Fu, Z., Wang, Y.: Stmtrack: Template-free visual tracking with space-time memory networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13774–13783 (2021) 5, 12, 13, 14
21. Gao, B., Spratling, M.W.: Robust template matching via hierarchical convolutional features from a shape biased CNN. arXiv:2007.15817 (2020) 7
22. Gao, P., Zhang, Q., Wang, F., Xiao, L., Fujita, H., Zhang, Y.: Learning reinforced attentional representation for end-to-end visual tracking. Information Sciences **517**, 52–67 (2020) 5
23. Gladh, S., Danelljan, M., Khan, F.S., Felsberg, M.: Deep motion features for visual tracking. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 1243–1248. IEEE (2016) 4
24. Guo, D., Shao, Y., Cui, Y., Wang, Z., Zhang, L., Shen, C.: Graph attention tracking. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 5
25. Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., Wang, S.: Learning dynamic siamese network for visual object tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1763–1771 (2017) 4
26. He, A., Luo, C., Tian, X., Zeng, W.: Towards a better match in siamese network based visual object tracker. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 0–0 (2018) 4
27. 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) 4
28. Huang, L., Zhao, X., Huang, K.: GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019) 4, 11, 13
29. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE transactions on pattern analysis and machine intelligence **34**(7), 1409–1422 (2011) 5
30. Kersten, D., Mamassian, P., Yuille, A.: Object perception as bayesian inference. Annu. Rev. Psychol. **55**, 271–304 (2004) 2
31. Kiani Galoogahi, H., Fagg, A., Huang, C., Ramanan, D., Lucey, S.: Need for speed: A benchmark for higher frame rate object tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1125–1134 (2017) 4, 11, 12
32. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Kämäräinen, J.K., Danelljan, M., Zajc, L.Č., Lukežič, A., Drbohlav, O., et al.: The eighth visual object tracking VOT2020 challenge results. In: European Conference on Computer Vision, pp. 547–601. Springer (2020) 4, 11, 13
33. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., ˇCehovin Zajc, L., Vojir, T., Bhat, G., Lukezic, A., Eldesokey, A., et al.: The sixth visual object tracking VOT2018 challenge results. In: Proceedings of the European Conference on Computer Vision Workshops, pp. 0–0 (2018) 13
34. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kamarainen, J.K., ˇCehovin Zajc, L., Drbohlav, O., Lukezic, A., Berg, A., et al.: The seventh visual object tracking VOT2019 challenge results. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp. 0–0 (2019) 13
35. 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 Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4282–4291 (2019) 4, 12, 14
36. Lochmann, T., Deneve, S.: Neural processing as causal inference. Current Opinion in Neurobiology **21**(5), 774–781 (2011) 2
37. Lochmann, T., Ernst, U.A., Deneve, S.: Perceptual inference predicts contextual modulations of sensory responses. Journal of Neuroscience **32**(12), 4179–4195 (2012) 2

38. Lukezic, A., Matas, J., Kristan, M.: D3S-a discriminative single shot segmentation tracker. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7133–7142 (2020) 4, 14
39. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Robust visual tracking via hierarchical convolutional features. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018) 4
40. Ma, Z., Wang, L., Zhang, H., Lu, W., Yin, J.: RPT: Learning point set representation for siamese visual tracking. arXiv:2008.03467 (2020) 4, 12, 14
41. Mangawati, A., Leesan, M., Aradhya, H.R., et al.: Object tracking algorithms for video surveillance applications. In: 2018 International Conference on Communication and Signal Processing (ICCSP), pp. 0667–0671. IEEE (2018) 2
42. Mbelwa, J.T., Zhao, Q., Wang, F.: Visual tracking tracker via object proposals and co-trained kernelized correlation filters. The Visual Computer **36**(6), 1173–1187 (2020) 4
43. Mondragón, I.F., Campoy, P., Martinez, C., Olivares-Méndez, M.A.: 3d pose estimation based on planar object tracking for UAVs control. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 35–41. Ieee (2010) 2
44. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for UAV tracking. In: European Conference on Computer Vision (ECCV), pp. 445–461. Springer (2016) 4, 11, 12
45. Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 300–317 (2018) 4, 11, 13
46. 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 (CVPR), pp. 4293–4302 (2016) 4
47. Pan, Z., Liu, S., Sangaiah, A.K., Muhammad, K.: Visual attention feature (VAF): a novel strategy for visual tracking based on cloud platform in intelligent surveillance systems. Journal of Parallel and Distributed Computing **120**, 182–194 (2018) 2
48. Prabhakar, G., Kailath, B., Natarajan, S., Kumar, R.: Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving. In: 2017 IEEE Region 10 Symposium (TENSYMP), pp. 1–6. IEEE (2017) 2
49. Rao, R.P., Ballard, D.H.: Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience **2**(1), 79–87 (1999) 2
50. Solbakken, L.L., Junge, S.: Online parts-based feature discovery using competitive activation neural networks. In: The 2011 International Joint Conference on Neural Networks, pp. 1466–1473. IEEE (2011) 8
51. Spratling, M.W.: Image segmentation using a sparse coding model of cortical area v1. IEEE transactions on image processing **22**(4), 1631–1643 (2012) 8
52. Spratling, M.W.: Classification using sparse representations: a biologically plausible approach. Biological cybernetics **108**(1), 61–73 (2014) 8
53. Spratling, M.W.: Predictive coding as a model of cognition. Cognitive Processing **17**(3), 279–305 (2016) 2
54. Spratling, M.W.: A hierarchical predictive coding model of object recognition in natural images. Cognitive Computation **9**(2), 151–167 (2017) 2
55. Spratling, M.W.: A review of predictive coding algorithms. Brain and Cognition **112**, 92–97 (2017) 2
56. Spratling, M.W.: Explaining away results in accurate and tolerant template matching. Pattern Recognition p. 107337 (2020) 2, 7, 8, 9, 17
57. Spratling, M.W., De Meyer, K., Kompass, R.: Unsupervised learning of overlapping image components using divisive input modulation. Computational intelligence and neuroscience **2009** (2009) 8
58. Tan, H., Zhang, X., Zhang, Z., Lan, L., Zhang, W., Luo, Z.: Nocal-siam: Refining visual features and response with advanced non-local blocks for real-time siamese tracking. IEEE Transactions on Image Processing (2021) 5, 13, 14, 15
59. Tarhan, M., Altuğ, E.: A catadioptric and pan-tilt-zoom camera pair object tracking system for UAVs. Journal of Intelligent & Robotic Systems **61**(1-4), 119–134 (2011) 2

60. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.H.S.: End-to-end representation learning for correlation filter based tracking. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 4
61. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. arXiv:1706.03762 (2017) 5
62. Voigtlaender, P., Luiten, J., Torr, P.H., Leibe, B.: Siam R-CNN: Visual tracking by re-detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6578–6588 (2020) 5, 12, 13, 14
63. Wang, J., He, Y.: Motion prediction in visual object tracking. arXiv:2007.01120 (2020) 4
64. Wang, N., Zhou, W., Wang, J., Li, H.: Transformer meets tracker: Exploiting temporal context for robust visual tracking. arXiv:2103.11681 (2021) 4, 5, 11, 12, 13, 14
65. 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) 4, 14
66. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp. 7794–7803 (2018) 5
67. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp. 2411–2418 (2013) 4, 10, 12
68. Xu, T., Feng, Z., Wu, X.J., Kittler, J.: Adaptive channel selection for robust visual object tracking with discriminative correlation filters. International Journal of Computer Vision **129(5)**, 1359–1375 (2021) 4
69. Xu, T., Feng, Z.H., Wu, X.J., Kittler, J.: Joint group feature selection and discriminative filter learning for robust visual object tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7950–7960 (2019) 4
70. Xu, T., Feng, Z.H., Wu, X.J., Kittler, J.: Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking. IEEE Transactions on Image Processing **28**(11), 5596–5609 (2019) 4
71. Xu, T., Feng, Z.H., Wu, X.J., Kittler, J.: Learning low-rank and sparse discriminative correlation filters for coarse-to-fine visual object tracking. IEEE Transactions on Circuits and Systems for Video Technology **30**(10), 3727–3739 (2019) 4
72. Xu, T., Feng, Z.H., Wu, X.J., Kittler, J.: AFAT: adaptive failure-aware tracker for robust visual object tracking. arXiv:2005.13708 (2020) 4
73. Xu, Y., Wang, Z., Li, Z., Yuan, Y., Yu, G.: SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines. In: The Association for the Advancement of Artificial Intelligence (AAAI), pp. 12549–12556 (2020) 4, 13, 14
74. Xuan, S., Li, S., Zhao, Z., Kou, L., Zhou, Z., Xia, G.S.: Siamese networks with distractor-reduction method for long-term visual object tracking. Pattern Recognition p. 107698 (2020) 5
75. Yan, B., Zhang, X., Wang, D., Lu, H., Yang, X.: Alpha-refine: Boosting tracking performance by precise bounding box estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5289–5298 (2021) 3, 4, 10, 11, 12, 13, 14, 17
76. Zhang, Z., Li, B., Hu, W., Peng, H.: Towards accurate pixel-wise object tracking by attention retrieval. arXiv:2008.02745 (2020) 4, 14
77. Zhang, Z., Peng, H.: Deeper and wider siamese networks for real-time visual tracking. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 4
78. Zhipeng, Z., Houwen, P., Jianlong, F., Bing, L., Weiming, H.: Ocean: Object-aware anchor-free tracking. In: European Conference on Computer Vision (2020) 4, 14
79. Zhou, J., Wang, P., Sun, H.: Discriminative and robust online learning for siamese visual tracking. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), vol. 34(07), pp. 13017–13024 (2020) 4, 12
80. Zhu, X.F., Wu, X.J., Xu, T., Feng, Z., Kittler, J.: Robust visual object tracking via adaptive attribute-aware discriminative correlation filters. IEEE Transactions on Multimedia (2021) 4

81. Zhu, X.F., Wu, X.J., Xu, T., Feng, Z.H., Kittler, J.: Complementary discriminative correlation filters based on collaborative representation for visual object tracking. IEEE Transactions on Circuits and Systems for Video Technology **31(2)**, 557–568 (2020) 4
82. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 101–117 (2018) 3, 4, 5, 17, 18

Bo Gao received a M.Sc. degree in Electrical engineering from Central South University, Changsha, China in 2018. He is currently pursuing the Ph.D. degree in Computer Science in the Department of Informatics, King's College London, London, UK. His research interests include object tracking, computer vision and deep learning.

Michael Spratling received a B.Eng. degree in Engineering Science from Loughborough University and M.Sc. and Ph.D. degrees in Artificial Intelligence and Neural Computation from the University of Edinburgh. He is currently Reader in Computational Neuroscience and Visual Cognition at the Department of Informatics, King's College London. His research is concerned with understanding the computational and neural mechanisms underlying visual perception.