Multiple Context Features in Siamese Networks for Visual Object Tracking

Henrique Morimitsu[0000−0001−9455−8571]

Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France
hmorimitsu@outlook.com

Abstract. Siamese networks have been successfully utilized to learn a robust matching function between pairs of images. Visual object tracking methods based on siamese networks have been gaining popularity recently due to their robustness and speed. However, existing siamese approaches are still unable to perform on par with the most accurate trackers. In this paper, we propose to extend the SiamFC tracker [1] to extract features at multiple context and semantic levels from very deep networks. We show that our approach effectively extracts complementary features for siamese matching from different layers, which provides a significant performance boost when fused. Experimental results on VOT and OTB datasets show that our multi-context tracker is comparable to the most accurate methods, while still being faster than most of them. In particular, we outperform several other state-of-the-art siamese methods.

Keywords: Object tracking · Siamese network · ResNet

1 Introduction

Visual object tracking consists of estimating the trajectory of an object along a continuous video sequence. Usually, only the first frame is annotated with a bounding box, which provides very limited information about the object to be tracked. In real situations, the target often undergoes complex transformations which cause its appearance to significantly change over time. Until recently, the majority of trackers tackled this challenge by constantly updating a classifier throughout the video [7, 9, 13]. In fact, when combined with deep network models, this strategy still produces the most accurate results on standard benchmarks [4, 26, 23]. However, updating the classifier online presents challenges of its own. Firstly, constantly updating a large model causes a significant drop in speed. Secondly, as the update depends on previous predictions, the classifier is prone to drift and contamination [6, 19, 31].

Lately, however, siamese networks have shown that compelling results can be achieved without updating the model [1, 10, 12]. Siamese trackers are trained on a large set of image pairs to learn a robust matching function that is able to re-identify the object even when its appearance changes significantly. Nonetheless, although they are usually fast, there is still a gap in accuracy when compared to the top-performing trackers.

In this paper, we show that this gap can be significantly decreased by collecting features containing different context and semantic levels from a deep network. Unlike traditional multi-layer tracking approaches which only exploit the different semantic levels [24, 28, 30], we extract features with multiple context levels by applying a crop on the feature maps, which we refer to as *multi-context features*. In the scope of this work, *context* refers to the amount of background that is included with the object. Figure 1b shows an example of an object with different context levels. Since the receptive field is different at each layer, cropping the maps allows each feature to collect information from different context sizes. We hypothesize, and show through experiments, that multi-context features are particularly suitable for siamese networks.

In the siamese formulation proposed in SiamFC [1], two images, an exemplar z and an instance x, are forwarded through two identical networks with shared weights, yielding features $\varphi(z)$ and $\varphi(x)$ respectively. When matching the features, $\varphi(z)$ can be interpreted as a filter to be applied over $\varphi(x)$ to produce a prediction. If we use multi-layer features, it is possible to obtain multiple filters $\varphi_l(z)$ and $\varphi_l(x)$. However, standard multi-layer features can only provide different global representations for the same image. On the other hand, as explained in §3.2, by considering *multi-context features*, filters from different layers can be more diverse, focusing on different regions of the image. As discussed in previous works [1, 10, 29], the amount of context can play a significant role in the tracking performance. And our proposed tracker, SiamMCF, allows to leverage it at multiple levels in a single pass.

The contributions of this paper are two-fold: (i) a novel extension to the siamese formulation which leverages multiple context and semantic levels in a single forward pass, and (ii) we demonstrate that the multi-context features provide a significant increase in performance when compared to standard multi-layer ones.

2 Related work

2.1 Siamese tracking

SINT [27] is one of the earliest siamese trackers that presented some really compelling results. It consists of a siamese network trained for matching patches of images. For the tracking stage, a patch of the first frame is matched to patches collected around the previous position. Although its results are still among the best siamese trackers, it is much slower than other approaches. GOTURN [12], on the other hand, is able to track at 100fps. It works by extracting deep features from two crops: one from the object and another from the area centered on the previous position. These features are concatenated and used to solve a regression problem to estimate the relative motion of the target relative to the previous frame. The high speed does compromise the performance, as its results are not as accurate as other siamese approaches.

SiamFC [1] is one of the most balanced options, as it presents one of the best compromises between accuracy and speed. The SiamFC tracker employs

a pair of AlexNets [18] with shared weights. A smaller exemplar image and a larger instance are forwarded to generate high-level features. By correlating the exemplar feature over every instance position, a spatial prediction map is obtained.

Several improvements [8, 10, 14, 29] were proposed over the initial SiamFC tracker. CFNet [29] proposes to include a trainable correlation filter layer on top of the siamese network. By introducing a differentiable solution to the deep correlation filter in the Fourier domain, the tracker can be efficiently trained end-to-end with gradient descent. DSiam [8] tackles the model updating problem in siamese networks. Two transformation terms are independently applied to both branches before the matching. The first term updates the model by encouraging it to be similar to the previous observation. The second one is used to suppress background activations in the current frame. EAST [14] proposes to speed-up SiamFC by trying to avoid forwarding the images until the last layer. For this, reinforcement learning is applied to train a classifier that decides at which layer forwarding can be stopped while still retaining a discriminative representation for the given image. SA-Siam [10] leverages appearance and semantic features for tracking. This is done by using two networks, a SiamFC and an AlexNet trained for classification. The authors show that the features obtained from each net are complementary and better results are obtained by combining their predictions.

2.2 Tracking with multi-layer representations

Multi-layer features have been applied to object tracking in different ways. Wang *et al.* [30] showed that different layers effectively produce complementary features for tracking. By leveraging information collected from different layers, tracking results were improved. Chi *et al.* [3] also exploited this property to obtain predictions from multiple layers. Some other methods [22, 28] have used deeper layers to first roughly estimate the target position and then project it to shallower layers. The rationale is that early layers provide less coarse features which can improve the detection accuracy. HDT[24] applies an adaptive hedge method to assign different confidence levels to each layer based on its previous results. C-COT [7] employs an implicit interpolation model to cast the feature maps into the continuous space. In this way, features from different layers with different sizes can be merged to train a correlation filter.

All of the previous approaches still use the whole feature maps for the predictions. Therefore, complementarity between layers is somewhat restricted, as all layers are global representations of the image and do not fully exploit information related to more localized patches. Some recent works [5, 21] have demonstrated that, by suppressing or masking the features maps, more robust representations can be obtained. In this work, we propose to combine multi-layer features with spatially constrained maps to obtain multi-context features. SA-Siam [10] has exploited this property to some extent by concatenating features from two layers and then cropping. However, that was applied to consecutive layers of an AlexNet [18], which are not very deep and too close to each other. SINT [27]

adopts a strategy to extract multi-layer features which is similar to ours. However, there are some important differences to be pointed out. Firstly, we apply cropping on the exemplar branch, in which the object is known, whereas SINT uses ROI pooling on the instance branch, in which the real object position is uncertain. Secondly, ROI pooling in SINT performs a rescaling (into a 7x7 region) with a max-pooling directly in the lower-resolution feature space, while we rescale the input image before forwarding and always crop a region of the same size, which is less prone to suffer from the negative effects of discretization. We show through experiments that our approach obtains significantly better results than other previous siamese approaches, including SINT and SiamFC-R [15], which also uses a very deep network as a backbone.

3 Our approach

3.1 Siamese tracker

In the standard SiamFC [1] formulation, two images are provided as inputs: the exemplar from the first frame z and the current tracking frame, the instance x. Let the prime symbol represent a crop operation over an image. The siamese network receives the cropped regions z' and x' which are then forwarded to produce the features $\varphi(z')$ and $\varphi(x')$ respectively. The feature $\varphi(z')$ is then used as a correlation filter over $\varphi(x')$, thus yielding a prediction map

$$g(x', z') = \varphi(x') \star \varphi(z'). \tag{1}$$

3.2 Siamese tracker with multi-context features

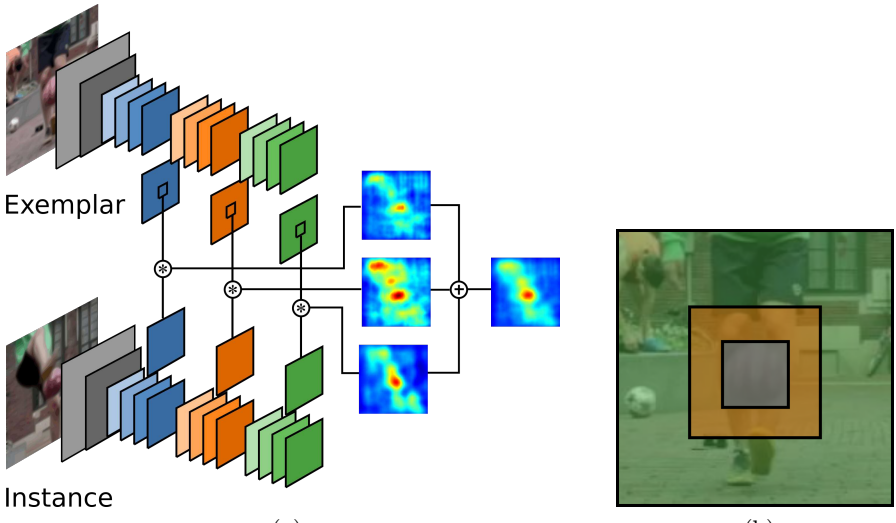
For our SiamMCF tracker, we adapt the prediction map function to work at different layers and extract features with different contexts from each layer. Figure 1 illustrates our proposed approach. The context amount is controlled by cropping the feature map. Since the receptive field at each layer is different, as long as the crop sizes in different layers are not proportional to the receptive field changes, we are able to extract features that consider different areas of the input image. In particular, we can extract features with different contexts by cropping regions of the same size from all the layers. Figure 1b shows the effective region corresponding to crops at different layers.

Given a set of selected layers $L = \{l\}$, prediction maps are estimated as

$$g_l(x', z') = \mathbb{1}\gamma_l \odot (\varphi_l(x') \star \varphi'_l(z')) + \mathbb{1}\beta_l, \tag{2}$$

where $\varphi_l(\cdot)$ represents the feature obtained by forwarding until layer l, and $\odot$ indicates element-wise multiplication. We also learn normalization parameters γ_l and β_l to stabilize the magnitude of the predictions. Notice that we use the cropped filter $\varphi'_l(z')$ to collect exemplars with different context sizes.

Since the backbone network in SiamFC is based on AlexNet [18], which is relatively shallow, extracting multi-level features does not provide very different



(a) (b)

Fig. 1: Illustration of our tracking framework. (a) Proposed network with multi-context features. (b) Receptive fields of different layers superposed over the image. Deeper layers encode larger contexts. Best viewed in colors.

representations. Therefore, we replace the backbone with a deeper network. In particular, we conduct experiments with a ResNet-50 [11]. The original ResNet, however, has a large output stride of 32, which is not ideal for the siamese formulation as both images are largely reduced. Therefore, we reduce the output stride to 8 by setting the convolution stride to 1 in blocks 2 and 3 of the ResNet, and by applying dilated convolutions [2].

It is important to mention that the original SiamFC is based on the fully-convolutional formulation [1]. This formulation ensures that the output features generated by the network commute with translation. Therefore, if the exemplar image is a crop of a region of the instance, then the exemplar output features will also correspond to a region of instance features. In other words, the exemplar image can be found in the instance simply by looking for the region with the maximum similarity. One important caveat is that this formulation can only hold as long as the employed network does not use padding operations, which severely restricts the choice of available architectures.

A ResNet, however, is very deep and requires padding. In fact, the receptive field in the last layer is usually larger than the input image, thus generating an asymmetry when processing images of different sizes (e.g. 127 and 255 for the exemplar and instance branches) which, in turn, break the fully-convolutional formulation. We hypothesize, and show by experiments, that the use of multi-context features alleviates this issue, by using images of the same size, and by extracting cropped intermediate features which: (i) are comparable due to same

size inputs, and (ii) also include features whose receptive field are smaller than the input (earlier layers).

We further modify the network by adding residual adaptation modules on top of each of the $|L|$ base layers from the backbone network. A residual adaptation module consists of an additional bottleneck residual unit [11] followed by a convolution. The residual unit has the same properties (number of channels, dilation rate, etc.) as the base ResNet layer it is connected to. The role of this module is to provide more capacity for the extracted features to adapt to the siamese matching at each layer and also to decrease the dimensionality for faster cross-correlation. We show experimentally that the addition of residual units for adaptation positively affects the results.

Final predictions are obtained by computing the average map:

$$g(x', z') = \frac{1}{|L|} \sum_{l \in L} g_l(x', z'). \tag{3}$$

3.3 Training

We compute an individual loss to each layer prediction $g_l(x', z')$. Let i indicate the index of the element (pixel) in a map. Then the loss of each prediction is the average of the logistic losses ℓ_l:

$$\mathcal{L}_l = \sum_i w(y_i) \ell_l(g_l(x'_i, z'_i; \theta), y_i), \tag{4}$$

where $w(y_i)$ is a weighting function applied on the labels y_i that leverages the imbalance between positive and negative samples. This weighting function is defined as:

$$w(y_i) = \frac{0.5 y_i}{n_{\text{pos}}} + \frac{0.5(1 - y_i)}{n_{\text{neg}}}, \tag{5}$$

where n_{pos} and n_{neg} are the number of positive and negative samples respectively.

The network is then trained with gradient descent to find the set of parameters θ that minimizes the global loss:

$$\theta^* = \arg\min_\theta \sum_{l \in L} \mathcal{L}_l(g_l(x'_i, z'_i; \theta), y_i) + \lambda \|\theta\|_2^2. \tag{6}$$

4 Experimental Results

4.1 Datasets and evaluation protocols

We evaluate our tracker on two widely-adopted public datasets: the visual object tracking (VOT) and the online tracking benchmark (OTB).

VOT. Both VOT16 and VOT17 [15–17] are composed of 60 sequences annotated with rotated bounding boxes. The standard evaluation criterion is focused on short-term tracking, where trackers are reinitialized whenever their IoU is zero. Trackers are ranked mainly according to three measures: Expected Average Overlap, Accuracy and Robustness. It also provides a normalized speed value (EFO) which can be used to compare tracker speeds disregarding the influence of the hardware, to some extent. (we refer to [16] for more details about the metrics).

OTB. We use two versions of the OTB dataset: OTB13 [32] and OTB15 [33]. The former contains 51 objects to be tracked, while the latter is a superset of OTB13 with 100 objects. The trackers are evaluated by two measures: precision and success. Precision estimates the average distance between the center of the predicted bounding box and the groundtruth. Success is used for obtaining the average Intersection-over-Union (IoU) of the predicted boxes. We use OTB13 for our ablation experiments, while OTB15 is kept for comparing with state-of-the-art trackers.

4.2 Implementation details

Network. Our backbone network is a ResNet [11] with 50 layers. We initialize its weights from a model trained on ImageNet [25] classification. As mentioned before, we decrease the network output stride from 32 to 8. In order to keep the input size compatible with the stride, we resize the input images to 248×248 pixels. In our formulation, both the exemplar and the instance images are of the same size and they include a large context, which is obtained by cropping an area 16 times larger than the object. The output features generated by the network have dimensions $31 \times 31 \times 64$. For the multi-context features, we crop the central 7×7 region from each of the feature maps $\varphi_l(z')$. Our set of chosen layers L is composed of the outputs of each of the 4 residual blocks of the ResNet.

Training. During training, the weights from the ResNet are frozen, and only the residual adaptation modules are trained. We briefly experimented with training ResNet layers as well, but we did not observe any noticeable improvement. The training follows the same protocol as in the SiamFC [1], by learning to match pairs of images collected from the ImageNet VID challenge. This dataset contains around 4000 sequences divided into 30 categories, which accounts for more than one million frames. One important point to notice is that, since ResNets use padded convolutions, the training targets must not always be centered, as it is done in SiamFC. Otherwise, the network will learn a positional bias. Therefore, we augment the training set with random cropping, as well as color distortion, horizontal flipping, and small resizing perturbations. The weights are optimized using gradient descent with a momentum term of 0.9. The learning rate is continually decayed exponentially from 10^{-3} to 10^{-6}. The network is trained during 50000 iterations with a mini-batch size of 8 pairs of images.

Testing. Tracking is conducted in the same manner as in the SiamFC [1]. Therefore, the matching is conducted independently at each frame and spatial consistency between frames is enforced by applying a Hann window over the prediction map. In order to obtain more precise predictions, we upsample the correlation output by a factor of 8 using bicubic interpolation. We handle scale changes by forwarding three images at different scales. For a fair comparison, all hyperparameters are kept the same as in SiamFC.

We implement our tracker using Python and Tensorflow 1.4. The experiments were conducted on a machine with an Intel Xeon E5 CPU and a GeForce GTX 1080Ti GPU. The average tracking speed during the experiments is around 20 frames per second. The code will be made available on `github.com/hmorimitsu/siam-mcf`.

4.3 Ablation study

We verify the contribution of each of our design choices by evaluating the results of different configurations on the OTB13 dataset. Our main interests were to verify the impacts of (i) replacing the AlexNet in SiamFC by a ResNet, (ii) using different layers from the ResNet for the matching, (iii) using large context inputs with late feature cropping, and (iv) including residual adaptation modules. For the third test, when large-context and cropping are not used, we input an exemplar image whose size is 120×120 pixels. This image also contains a reduced context size, corresponding to an area four times larger than the object, which is the same setting used in SiamFC. For the fourth test, if residual adaptation is not used, we add and train only a single convolutional layer on top of the ResNet outputs. Table 1 summarizes our results.

The last row corresponds to the result obtained by the baseline SiamFC. The results show that simply replacing the backbone with a ResNet actually generates worse results. This can be explained by the violation of the fully-convolutional formulation discussed in §3.2. Even by considering multi-layer features, the performance is only on par with the baseline. However, as illustrated by the results in the bottom part of the table, multi-context features obtained with feature cropping from multiple layers produce noticeably better results. In fact, even when applied to some layers individually, the results are already better than the baseline. However, we see that by combining it with multiple layers we have significantly better performance. It is interesting to remark that, although using L4 by itself usually leads to worse results, removing it from the multi-features set actually generates slightly worse results. One reason is that, in sequences such as Ironman, MotorRolling, and Skating1, L4 is actually better than other layers. We observe a similar behavior when comparing L123 with L1234, thus showing that L4 predictions are beneficial to the model. Lastly, we observe that dropping the residual adaptation does decrease the results, thus demonstrating its contribution.

We select the model with multi-context features and residual adaptation module, which generated the best results, as our SiamMCF to perform the experiments against the state-of-the-art methods.

Table 1: Ablation results on OTB13 dataset. L1 – L4 indicates that features from those levels are being used for matching.

ResNet	L1	L2	L3	L4	Feat. crop	Res adapt.	IoU	Prec.
✓	✓					✓	0.517	0.671
✓		✓				✓	0.549	0.704
✓			✓			✓	0.584	0.782
✓				✓		✓	0.506	0.718
✓	✓	✓	✓	✓		✓	0.612	0.801
✓	✓	✓	✓	✓			0.577	0.748
✓	✓				✓	✓	0.592	0.775
✓		✓			✓	✓	0.654	0.846
✓			✓		✓	✓	0.654	0.871
✓				✓	✓	✓	0.535	0.740
✓	✓	✓	✓		✓	✓	0.676	0.876
✓	✓	✓	✓	✓	✓	✓	**0.692**	**0.894**
✓	✓	✓	✓	✓	✓		0.688	0.879
baseline SiamFC							0.606	0.807

Table 2: Results on the VOT16 dataset. The arrows indicate whether higher or lower values are better.

Tracker	SiamMCF	SiamRPN	C-COT	TCNN	SSAT	SA-Siam	SiamFC-R	SiamFC-A
EAO ↑	**0.361**	0.344	0.331	0.325	0.321	0.291	0.277	0.235
Acc. ↑	**0.58**	0.56	0.54	0.55	0.58	0.54	0.55	0.53
Rob. ↓	1.05	1.08	0.89	**0.83**	1.05	1.08	1.36	1.91
EFO ↑	5.5	**23.3**	0.5	1.0	0.5	< 9	5.4	9.2

4.4 Comparison with the start-of-the-art

We validate the performance of our tracker by comparing its results with some state-of-the-art trackers. We selected some of the currently best performing trackers in general, as well as other recent siamese proposals. We evaluate our results on three datasets: VOT16, VOT17, and OTB15.

VOT16. We compare our results using SiamMCF on VOT 2016 with the best contenders in the competition (C-COT, TCNN, SSAT, MLDF). We also include the results of other trackers, including SA-Siam [10] and SiamRPN [20], two recent siamese trackers, and SiamFC-R[15], a SiamFC modified to use ResNet as a backbone. The results are summarized in Table 2.

We can see that SiamMCF outperforms all compared trackers, including the best tracker in the competition, C-COT, and recent siamese methods SiamRPN and SA-Siam. On the other hand, we still cannot obtain better robustness than

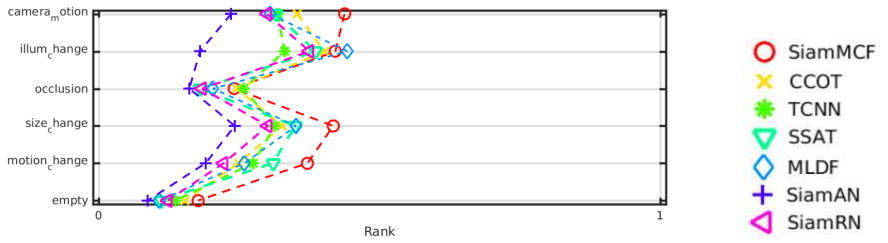


Fig. 2: EAO ranking on VOT16 according to sequence attributes. Each row corresponds to an attribute. The horizontal axis shows the EAO according to the corresponding attribute. Our SiamMCF obtain the best results most of the time.

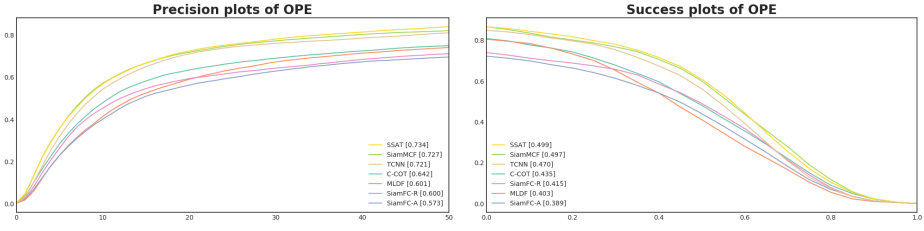


Fig. 3: OPE results on the VOT16 dataset.

the methods using online updating, although we outperform all siamese entries. By analyzing the ranking results in Figure 2, we see that occlusion is the main reason for the drop in performance. This result is understandable, as in such situation, the tracker tends to present higher activations in the surrounding area than in the occluded region, thus causing drifting. It is important to notice that the contributions of SiamMCF are orthogonal to siamese updating strategies, for example, as proposed by DSiam [8]. Therefore, it is possible that even better performance could be obtained by applying those updating strategies to our tracker.

We also compare our tracker on the unsupervised setting of the VOT benchmark. Different from the standard settings, the trackers are not reinitialized after they drift away from the target. This evaluation focuses on longer-term tracking, as it penalizes more heavily trackers which are unable to recover from a temporary target loss. Figure 3 shows the precision and success plots of the One-Pass Evaluation (OPE) on the VOT16 dataset. We can see that SiamMCF also achieves state-of-the-art results on this test, being very close to the best method SSAT.

VOT17. As in the previous benchmark, we also select the top trackers from the competition (LSART, CFWCR, CFCF, ECO) and siamese trackers (SiamDCF, SA-Siam, SiamFC). From the results in Table 3 we see that our tracker also

Table 3: Results on the VOT17 dataset. The arrows indicate whether higher or lower values are better.

Tracker	SiamMCF	LSART	CFWCR	CFCF	ECO	SiamDCF	SA-Siam	SiamFC
EAO ↑	0.304	**0.323**	0.303	0.286	0.280	0.249	0.236	0.188
Acc. ↑	**0.53**	0.49	0.48	0.51	0.48	0.50	0.50	0.50
Rob. ↓	1.31	**0.94**	1.21	1.17	1.12	1.87	-	2.05

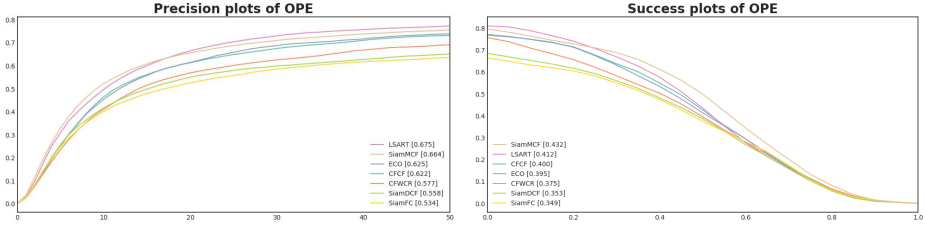


Fig. 4: OPE results on the VOT17 dataset.

performs very favorably on VOT17 as well, being seconded only by LSART, while retaining the highest accuracy. Once again we outperform all other siamese trackers.

The unsupervised results shown in Figure 4 are also encouraging. In this dataset, SiamMCF actually outperforms all other trackers when considering the Intersection over Union metric (success plots), while being close to the best method in terms of the center distance of the predictions (precision plots).

OTB15. We further verify the results of SiamMCF on OTB15. We show comparative results against state-of-the-art trackers that use multi-layer features (ECO [4], C-COT [7], HDT [24]) and siamese networks (SINT+ [27], SiamFC [1], CFNet [29]). Once again we outperform other siamese proposals while approaching the other state-of-the-art methods, which rely on online updating. It is worth noticing that ECO adopts different hyperparameters for OTB and VOT datasets, whereas we keep them fixed for all evaluations. Particularly, we kept SiamFC parameters for fair comparison, thus it is possible that a further improvement could be obtained with a careful hyperparameter search.

We also show the results on some more specific attributes in Figure 6. Similarly to what was observed on VOT, videos containing occluded or out-of-view objects are responsible for the largest differences in performance. On the other hand, our tracker performs remarkably well on low-resolution videos. This seems to be a feature of trackers based on SiamFC, as both SiamFC itself and CFNet also perform relatively better in this type of sequence. Some qualitative results are displayed in Figure 7.

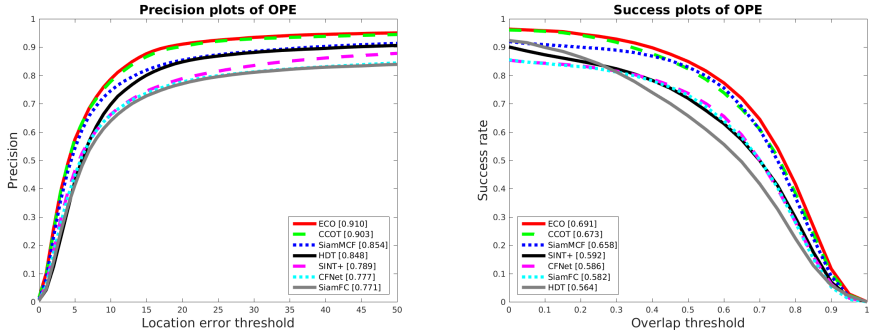


Fig. 5: Results on the OTB15 dataset.

We can see that SiamMCF is quite robust to diverse challenging situations, including change of lighting, rotation and scale changes. From the results of the fourth sequence, we see that the use of deeper networks provide additional robustness to rotation, as both our method and HDT show good results. Nonetheless, we observe that our proposal is still overall more robust than HDT, correctly tracking the target in sequences 2 and 3. The qualitative results also confirm that our approach is more robust than the SiamFC baseline, as it works correctly in many sequences where SiamFC loses the target. We also verify that we are able to better handle some sequences where the top performer ECO has difficulties.

The last two sequences present some failure cases for our tracker. We can see that sometimes when the target appearance changes significantly, or if fast motion and blur happen, we are still unable to keep tracking the target. Occlusion also presents difficulties, which may cause the tracker to drift away from the target.

5 Summary

This paper proposed to extend SiamFC to exploit multi-context features in visual object tracking, which is obtained by applying cropping on features maps of different layers. In this way, each layer contributes not only with different semantic levels, but also focus on regions of different sizes of the input image. We showed that by incorporating these features into a deep siamese network tracker we are able to obtain outstanding results in short-term tracking, by outperforming almost all other methods on the newest VOT benchmarks. We are also able to outperform the state-of-the-art siamese trackers on OTB while getting close to the most accurate methods. Even with the use of multi-context features and deep networks, our tracker remains faster than many of the top-performing methods, running at almost real-time speeds.

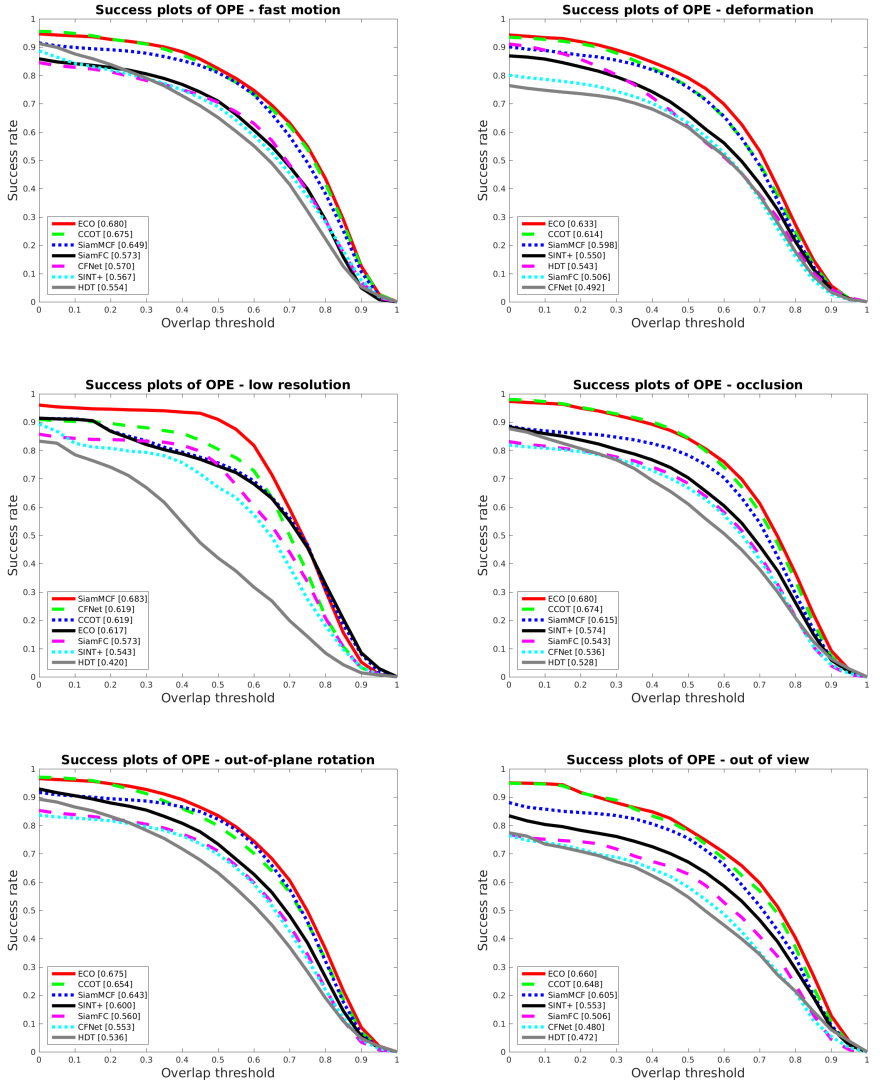


Fig. 6: OTB15 success plots for different attributes.

Acknowledgements

This work was supported in part by Indo-French project EVEREST (no. 5302-1) funded by CEFIPRA. I offer my most sincere regards to Dr. Karteek Alahari and Dr. Cordelia Schmid for providing invaluable insights and support for concluding this project. I also thank Rafael Eller Cruz for providing comments to improve this manuscript.

Fig. 7: Qualitative results on sequences from OTB15 using the selected trackers.

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: ECCV. pp. 850–865 (2016)

2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE TPAMI **40**(4), 834–848 (2018)
3. Chi, Z., Li, H., Lu, H., Yang, M.H.: Dual deep network for visual tracking. IEEE TIP **26**(4), 2005–2015 (2017)
4. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ECO: efficient convolution operators for tracking. In: CVPR. pp. 6638–6646 (2017)
5. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: ICCV. pp. 4310–4318 (2015)
6. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking. In: CVPR. pp. 1430–1438 (2016)
7. Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M.: Beyond correlation filters: learning continuous convolution operators for visual tracking. In: ECCV. pp. 472–488 (2016)
8. Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., Wang, S.: Learning dynamic siamese network for visual object tracking. In: ICCV. pp. 1763–1771 (2017)
9. Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.M., Hicks, S.L., Torr, P.H.S.: Struck: structured output tracking with kernels. IEEE TPAMI **38**(10), 2096–2109 (2016)
10. He, A., Luo, C., Tian, X., Zeng, W.: A twofold siamese network for real-time object tracking. In: CVPR. pp. 4834–4843 (2018)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
12. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 fps with deep regression networks. In: ECCV. pp. 749–765 (2016)
13. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE TPAMI **37**(3), 583–596 (2015)
14. Huang, C., Lucey, S., Ramanan, D.: Learning policies for adaptive tracking with deep feature cascades. In: ICCV. pp. 105–114 (2017)
15. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Čehovin, L., Vojíř, T., Gustav, H., et al.: The visual object tracking VOT2016 challenge results. In: Visual Object Tracking Workshop (2016)
16. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Čehovin, L., Vojíř, T., Gustav, H., et al.: The visual object tracking VOT2017 challenge results. In: Visual Object Tracking Workshop (2017)
17. Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Čehovin, L.: A novel performance evaluation methodology for single-target trackers. IEEE TPAMI **38**(11), 2137–2155 (2016)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1097–1105 (2012)
19. Kwon, J., Timofte, R., Van Gool, L.: Leveraging observation uncertainty for robust visual tracking. Computer Vision and Image Understanding **158**, 62–71 (2017)
20. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: CVPR. pp. 8971–8980 (2018)
21. Lukezic, A., Vojír, T., Zajc, L.C., Matas, J., Kristan, M.: Discriminative correlation filter with channel and spatial reliability. In: CVPR. vol. 8, pp. 6309–6318 (2017)
22. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: ICCV. pp. 3074–3082 (2015)
23. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: CVPR. pp. 4293–4302 (2016)

24. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., Yang, M.H.: Hedged deep tracking. In: CVPR. pp. 4303–4311 (2016)
25. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
26. Sun, C., Wang, D., Lu, H., Yang, M.H.: Learning spatial-aware regressions for visual tracking. In: CVPR. pp. 8962–8970 (2018)
27. Tao, R., Gavves, E., Smeulders, A.W.M.: Siamese instance search for tracking. In: CVPR. pp. 1420–1429 (2016)
28. Teng, Z., Xing, J., Wang, Q., Lang, C., Feng, S., Jin, Y.: Robust object tracking based on temporal and spatial deep networks. In: CVPR. pp. 1144–1153 (2017)
29. Valmadre, J., Bertinetto, L., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: End-to-end representation learning for correlation filter based tracking. In: CVPR. pp. 5000–5008 (2017)
30. Wang, L., Ouyang, W., Wang, X., Lu, H.: Visual tracking with fully convolutional networks. In: ICCV. pp. 3119–3127 (2015)
31. Wang, N., Shi, J., Yeung, D.Y., Jia, J.: Understanding and diagnosing visual tracking systems. In: ICCV. pp. 3101–3109 (2015)
32. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: CVPR. pp. 2411–2418 (2013)
33. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE TPAMI **37**(9), 1834–1848 (2015)