Relation-aware Multiple Attention Siamese Networks for Robust Visual Tracking

Fangyi Zhang[1,2]
fangyi.zhang@vipl.ict.ac.cn
Bingpeng Ma[2]
bpma@ucas.ac.cn
Hong Chang[1,2]
changhong@ict.ac.cn
Shiguang Shan[1,2,3]
sgshan@ict.ac.cn
Xilin Chen[1,2]
xlchen@ict.ac.cn

[1] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China

[2] University of Chinese Academy of Sciences, Beijing, 100049, China

[3] CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai, 200031, China

Abstract

Partial occlusion is a challenging problem in visual object tracking. Neither Siamese network based trackers nor conventional part-based trackers can address this problem successfully. In this paper, inspired by the fact that attentions can make the model focus on the most salient regions of an image, we propose a new method named Relation-aware Multiple Attention (RMA) to address the partial occlusion problem. In the RMA module, part features generated from a set of attention maps can represent the discriminative parts of the target and ignore the occluded ones. Meanwhile, an attention regularization term is proposed to force the multiple attention maps to localize diverse local patterns. Besides, we incorporate relation-aware compensation to adaptively aggregate and distribute part features to capture the semantic dependency among them. We integrate the RMA module into Siamese matching networks and verify the superior performance of the RMA-Siam tracker on five visual tracking benchmarks, including VOT-2016, VOT-2017, LaSOT, OTB-2015 and TrackingNet.

1 Introduction

Visual object tracking is a fundamental problem in computer vision. Given the bounding box of the target in the first frame, visual trackers are required to not only localize the object position, but also decide its size in the following frames. Although great progress has been made over the last decades, building a robust tracker with high tracking speed is still a challenging task, especially at the presence of partial occlusion and deformation.

Recently, Siamese network based trackers [2, 14, 22, 40, 41, 49] have drawn much attention which achieve high tracking speed by avoiding online training. These Siamese trackers learn a similarity between holistic representations of the target in the template and the search

region. Unfortunately, only making use of the holistic model for target representation may easily miss fine-grained visual cues [24] and lead to degraded tracking results at the presence of partial occlusion and deformation [9, 46]. Traditional part-based trackers [11, 26, 27, 46] address this problem by explicitly dividing object with predefined grids into multiple parts and then learning correlation filters for each part. However, the target may not occupy all the grids. Moreover, the shape of different kinds of the target varies a lot. A fixed grid decomposition is suboptimal.

In this paper, we propose Relation-aware Multiple Attention (RMA) to handle the above problems. Instead of directly using a predefined decomposition, we incorporate multiple spatial attentions into Siamese networks to make the model automatically localize discriminative parts of the target. Specifically, the RMA module generates a set of attention maps that represent the most salient parts of the target. Meanwhile, an attention regularization term is utilized to remove the redundancy of the learned attention maps, forcing them to focus on diverse parts of the target. Therefore when the target is partially occluded, our multiple attention module can adaptively detect the visible parts while ignoring the occluded parts. Moreover, relation-aware semantic compensation is proposed to aggregate and distribute the local cues according to the correlation of each other, so that the local parts can capture the semantic dependency of each other. In this way, drifting to background with the similar local pattern can be greatly alleviated. We compare our method with Siamese based trackers on VOT-2016 [19], VOT-2017 [20], LaSOT [12], OTB-2015[45] and TrackingNet [29]. Experimental results show the superiority of our proposed method.

2 Related Work

Siamese Trackers. Visual object tracking can be viewed as a similarity matching problem. By comparing the target image patch with the candidate patches in a search region, we can track the object to the location where the highest similarity score is obtained. Similarity learning with deep convolutional neural networks is typically performed using Siamese architectures. The SINT [40] tracker trained a Siamese architecture to learn a metric for target matching. GOTURN [16] compared feature maps of the target and search region to find the target object with concatenation operation. SiamFC [2] brought cross correlation into a fully convolutional network with increased tracking accuracy and high tracking speed. DSiam [14] effectively online learned target appearance variation and suppressed background with a fast transformation learning model. The CFNet tracker [41] preformed online update of the Siamese network by integrating correlation filters into the network. SiamRPN [22] incorporated region proposal network to localize the target and estimate its size, avoiding exhaust search across different scales and significantly boosting the tracking speed. DaSiamRPN [49] improved the discrimination of the network with semantic negative pairs. However, all above methods encode the target in its entirety which leads to degraded tracking results in partial occlusion. Our method can greatly alleviate this problem by introducing relation-aware multiple attentions into the Siamese networks.

Part-based Trackers. Part-based trackers try to exploit object part information and achieve promising performance. Liu *et al*. [28] proposed to track the target based on multiple object parts with multiple independent correlation filters. Fan *et al*. [11] utilized motion models to preserve the inner structure of object parts. Du *et al*. [9] exploited high-order geometric relations among multiple parts of the target with hypergraph learning. However, the above

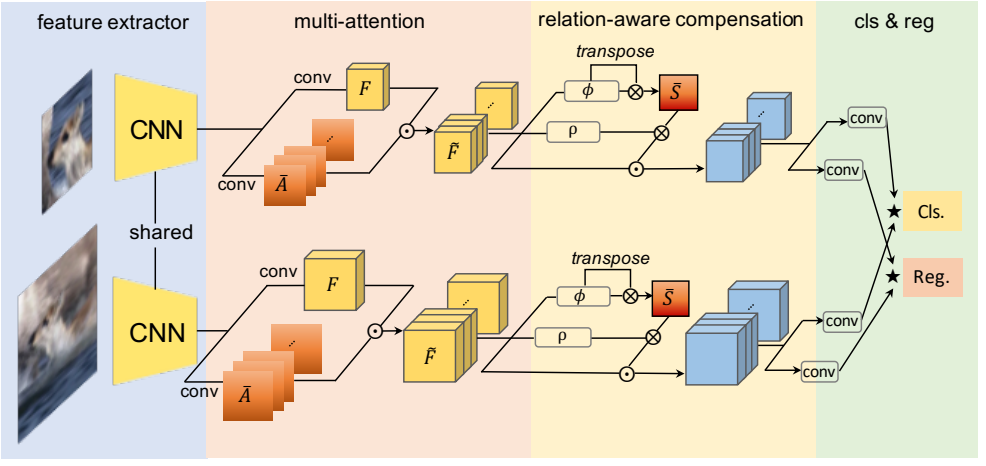

Figure 1: The overview of our proposed method. $\odot$ is the element-wise multiplication. $\otimes$ is the matrix multiplication. $\star$ is the cross-correlation operation.

methods use predefined grids of the target. The target may not occupy all the grids, a fixed grid decomposition is suboptimal. The most similar work to ours is StructSiam [48]. It simultaneously considered the local patterns of the target and their structural relationship with mean-field approximation algorithm. Our method is different from StructSiam in two aspects. First, our method learns multiple attention maps to automatically discover diverse local patterns in the spatial domain. Second, our method uses relation-aware compensation to enhance the semantic representation of the part feature maps, instead of using iterative mean-field approximation which costs much time.

Attention Mechanisms. Attention mechanisms are widely used in computer vision tasks, such as classification [17, 44], person re-identification [24], image synthesis [47], and action recognition [3, 43]. As for single object tracking problems, the FlowTrack [50] aggregated temporal features with temporal attention. The RASNet [42] integrated general attention, residual attention and channel attention to learn target specific representation. The ACFN [4] selected the tracking submodules with an attention network. The HART [33] utilized spatial attention to single out the tracked object and appearance attention to suppress distractors. The DAT [32] treated the gradient with respect to the image as an attention which helped the classifier attend to target regions. In contrast to the above methods, which only focus on the global view of the object, our method considers the object parts to improve the discrimination of the learned features.

3 The Proposed Approach

In this section, we will elaborate on the proposed method. Figure 1 illustrates the pipeline of our tracking algorithm. Given the template and a search image region, first we extract the features with a convolutional network. Second, the multiple spatial attention module is used to generate a set of diverse part features, each corresponding to a specific salient part of the target. Third, the part features are further enhanced by adaptive aggregation and distribution with relation-aware semantic compensation. Finally, we do classification and regression with

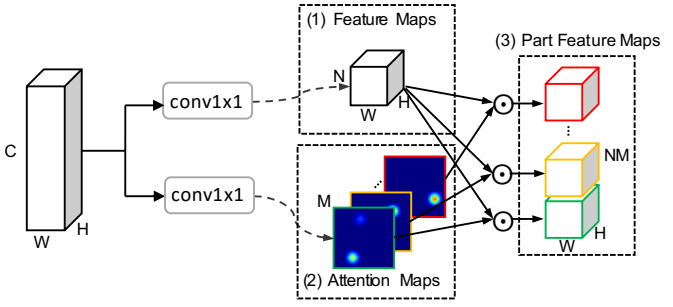

Figure 2: The multiple attention module. It first generates a compact feature map (1) and multiple attention feature maps (2) respectively with two 1×1 *conv*. The attention maps are normalized using softmax and element-wisely multiplied with the feature map to generate the part feature maps (3).

cross-correlation between the learned features of the template and the search region.

3.1 Multiple Spatial Attentions

Discriminative part features are critical when the target is occluded or has large deformation. Matching the global representations of the target between the template and the search region is not reliable [9, 46]. We propose the multiple spatial attention module, as shown in Figure 2, to automatically discover discriminative target parts instead of using predefined spatial decomposition of the target.

Multiple Attentions. We generate a feature map, $\mathbf{F} \in \mathbb{R}^{N \times H \times W}$, and a set of attention maps, $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_M\}$ where $\mathbf{a}_i \in \mathbb{R}^{H \times W}$, with two convolutional operations from the extracted feature maps. N and M are predefined positive integers, which decide the dimension of the embedding space and the number of attention maps, respectively. In our experiments, we set $N = 32$ and $M = 8$. Each attention map $\mathbf{a}_i$, which corresponds to a specific object part, is normalized using softmax of the responses:

$$\bar{\mathbf{a}}_i = \frac{\exp(\mathbf{a}_i)}{\sum_{j=1}^{H}\sum_{k=1}^{W}\exp(\mathbf{a}_{i,j,k})}, \quad i = 1, \ldots, M. \tag{1}$$

We element-wisely multiply the normalized attention map $\bar{\mathbf{a}}_i$ with the feature map $\mathbf{F}$ to generate the i^{th} part feature (we *rep*licate $\bar{\mathbf{a}}_i$ for N times to match the size of $\mathbf{F}$):

$$\tilde{\mathbf{F}}_i = rep(\bar{\mathbf{a}}_i) \odot \mathbf{F}, \quad i = 1, \ldots, M. \tag{2}$$

The final feature $\tilde{\mathbf{F}} = [\tilde{\mathbf{F}}_1; \tilde{\mathbf{F}}_2; \ldots; \tilde{\mathbf{F}}_M] \in \mathbb{R}^{NM \times H \times W}$ is concatenated by these part features.

Attention Regularization. Directly training without any supervision, the multiple spatial attentions will easily degenerate to locate on the same most salient part of the target. Inspired by RePr [31] which measures the inter-filter orthogonality with an orthogonal term, we use this term as a regularization to make different attention maps focus on diverse object parts. To this end, we reformulate the normalized attention maps as $\bar{\mathbf{A}} = [\bar{\mathbf{a}}'_i, \bar{\mathbf{a}}'_i, \ldots, \bar{\mathbf{a}}'_M] \in \mathbb{R}^{HW \times M}$,

Figure 3: Visualization of the learned multiple attention maps. The first and second row are the attention maps of *MotorRolling* and *Tiger1* respectively. Different columns represent different attention maps.

where $\bar{\mathbf{a}}'_i \in \mathbb{R}^{HW}$ is the vectorized attention map of $\bar{\mathbf{a}}_i$. The orthogonal loss is defined in the equation below:

$$L_{\text{orth}} = ||\mathbf{O}||_2 = ||\bar{\mathbf{A}}^T\bar{\mathbf{A}} - \mathbf{I}||_2, \tag{3}$$

where $\mathbf{O} \in \mathbb{R}^{M\times M}$ represents the orthogonality of the attention maps, $\mathbf{I} \in \mathbb{R}^{M\times M}$ is the identity matrix. Off-diagonal elements in a row of $\mathbf{O}$ denote the orthogonality of one attention map with all other attention maps. By minimizing the orthogonal loss, the attention maps are constrained to focus on diverse object parts. The learned multiple attention maps for some example images are illustrated in Figure 3.

3.2 Semantic Relation-aware Compensation

Matching local patterns between the template and the search region can easily cause the tracker drifting to background clutter which has similar local patterns. To alleviate this problem, relations among different part features must be taken into consideration so that part features not only represent the most salient object parts, but also capture the semantic dependency among them.

In order to capture the complex relations between part features, we calculate the correlation between different part features. We define $\mathbf{S} \in \mathbb{R}^{M\times M}$ as the affinity matrix:

$$\mathbf{S} = \phi(\tilde{\mathbf{F}})^T\phi(\tilde{\mathbf{F}}), \tag{4}$$

where $\phi(\tilde{\mathbf{F}}) \in \mathbb{R}^{N\times M}$ is the embedding function implemented by a 1×1 group convolutional layer followed by global average pooling and reshape operation. It transforms the original part features into a compact representation. The number of groups is same as the number of object parts. We use $\mathbf{S}_i$ to denote the i^{th} column of $\mathbf{S}$. It represents the similarity between the i^{th} part feature and the other part features. We normalize $\mathbf{S}_i$ with softmax function $\bar{\mathbf{S}}_i = \text{softmax}(\mathbf{S}_i)$.

Besides, we transform the original part features with function $\rho(\tilde{\mathbf{F}}) \in \mathbb{R}^{M\times HW}$ to an embedding space. The function ρ is another 1×1 group convolutional layer after with reshape operation. The number of groups is M. Then, the transformed part features are adaptively aggregated according to the learned correlation $\mathbf{S}$. Formally, the aggregated feature that will be assigned to the i^{th} part is computed as:

$$\mathbf{D}_i = \rho(\tilde{\mathbf{F}})^T\bar{\mathbf{S}}_i, \tag{5}$$

where $\mathbf{D}_i \in \mathbb{R}^{HW}$. We reshape $\mathbf{D}_i$ to 2D feature map in $\mathbb{R}^{H\times W}$ and combine it with the original i^{th} part feature map $\tilde{\mathbf{F}}_i$ using element-wise multiplication. In this way, each part feature can not only represent itself but also convey the complex relations with other parts.

3.3 Training and Tracking

Training. The training of our method is performed on image pairs, each of which is sampled from the same video sequence with a random interval. We use the following multi-task loss to end-to-end train our networks.

$$L = \sum_i L_{cls}(c_i, c_i^*) + \alpha \sum_i L_{loc}(l_i, l_i^*) + \gamma L_{orth}, \tag{6}$$

where L_{cls} is the softmax classification loss, L_{loc} is the smooth L_1 loss, and L_{orth} is defined in Eq. 3. α and γ are predefined weights to balance the three loss terms. In practice, we use $\alpha = 1.2$ and $\gamma = 0.01$. c_i and c_i^* are the predicted score and the label of the i^{th} anchor, respectively. l_i and l_i^* are the predicted offset and the offset between the i^{th} anchor and the corresponding groundtruth box. Following [35], let A_x, A_y, A_w, A_h denote the center coordinates, the width and height of the i^{th} anchor, and T_x, T_y, T_w, T_h denote those of the corresponding groundtruth box. Then $l_i^* = [l_{i(x)}^*, l_{i(y)}^*, l_{i(w)}^*, l_{i(h)}^*]$ is defined as follows:

$$l_{i(x)}^* = \frac{T_x - A_x}{A_w}, \quad l_{i(y)}^* = \frac{T_y - A_y}{A_h}, \quad l_{i(w)}^* = \ln \frac{T_w}{A_w}, \quad l_{i(h)}^* = \ln \frac{T_h}{A_h}. \tag{7}$$

Tracking. The setting of our RMA-Siam tracker is the same as the SiamPRN[22]. We extract features of the template and the search region using the same CNN. Then we find the most salient parts of the target with multiple attentions and enhance the part feature maps with our relation-aware compensation module. We do classification and regression with the enhanced part features in the template and the search region with cross-correlation. Cosine window is utilized to rescore the predicted classification score. The final box is determined by the maximal reweighted score. We also use linear interpolation to smooth the predicted size of the target.

4 Experiments

4.1 Implementation Details

We implement the proposed tracker with PyTorch 0.4.1 on a server with GTX-1080Ti GPU and Intel Xeon 2.2GHz CPU. The average tracking speed is **182fps** which drops 21fps compared with the baseline tracker SiamRPN running at the same environment. The backbone network of our architecture is modified AlexNet [21] which removes padding in all convolutional layers. We use one scale anchor with five aspect ratios which are $[0.33, 0.5, 1, 2, 3]$. We train our network on ImageNet VID [36] and YoutobeBB [34] dataset which is the same datasets used in SiamRPN for fair comparison. In both training and testing, we use single scale images with 127 pixels for template patches and 255 pixels for search regions. Our model is trained with stochastic gradient descent with momentum of 0.9. The whole network is end-to-end trained with 50 epochs with weight decay 10^{-4}. The initial learning rate is 10^{-2} and exponentially decayed to 10^{-5}.

4.2 Comparison with State-of-the-Arts

VOT-2016 Dataset. VOT-2016 dataset [19] consists of 60 challenging videos, where each sequence is per-frame annotated by five visual attributes, and the bounding box is generated

Tracker	SiamAN [2]	SA-Siam [15]	Staple [1]	DeepSRDCF [6]	ECO-HC [8]	C-COT [7]	SiamRPN [22]	RMA-Siam
A(↑)	0.54	0.54	0.54	0.56	0.54	0.54	**0.56**	**0.62**
Fail.(↓)	1.65	1.08	1.35	1.0	1.19	**0.85**	1.08	**0.95**
EAO(↑)	0.232	0.291	0.295	0.318	0.322	0.331	**0.344**	**0.382**

Table 1: Comparisions on VOT-2016 [19]. The best two results are highlighted in **red** and **blue**, respectively.

Tracker	GMDNetN [30]	SiamFC [2]	SA-Siam [15]	SiamRPN [22]	C-COT [7]	CFCF [13]	ECO [8]	LSART [39]	RMA-Siam
A(↑)	**0.513**	0.503	0.500	0.490	0.494	0.509	0.484	0.495	**0.583**
R(↓)	0.696	0.585	0.459	0.460	0.318	0.281	**0.276**	**0.218**	0.370
EAO(↑)	0.157	0.188	0.236	0.244	0.267	0.286	0.280	**0.323**	**0.311**

Table 2: Comparisions on VOT-2017 [20]. The best two results are highlighted in **red** and **blue**, respectively.

from the pixel-wise segmentation of the tracked object. In Table 1, we compare our tracker in terms of Expected Average Overlap (EAO), Accuracy (A), and Failure (Fail.) with some top-ranked trackers in the VOT-2016 benchmark. Our tracker achieves the best results with respect to EAO and A measurements, and outperforms the baseline tracker SiamRPN. We get 11% relative gains on EAO compared with SiamRPN. The most robust tracker is C-COT [7] as it makes use of online learning while our tracker relies solely on the first frame.

VOT-2017 Dataset. The VOT-2017 dataset [20] drops 10 videos which are easy to track compared with the VOT-2016 benchmark and adds another 10 more challenging videos. We compare our tracker with 8 top-ranked trackers on VOT-2017. Following the evaluation protocol of VOT-2017, we adopt EAO, Accuracy and Robustness (R) to evaluate different trackers. The detailed comparisons are reported in Table 2. The proposed tracker achieves the top-ranked performance with respect to EAO and A. Compared with the baseline tracker SiamRPN, we achieve 27.5% relative gains on EAO. In addition, compared with the top-performance of LSART [39], our tracker shows competitive performance.

LaSOT Dataset. The LaSOT dataset[12] provides a large-scale, high-quality dense annotations with 1400 videos in total. We follow the protocol II which uses 280 testing videos to evaluate our tracker with normalized precision plots and success plots. Figure 4 reports the overall performances of our tracker. We compare our tracker with 9 top performance approaches, including MDNet [30], VITAL [38], SiamFC [2], StructSiam [48], DSiam[14], ECO[8], SINT[40], STRCF[23], and DaSiamRPN [49]. Our tracker achieves relative gains of 5.4% on normalized precision plots and 4.6% on success plots compared the best performance tracker DaSiamRPN which is the enhanced version of SiamRPN.

Further, we analyze our tracker with respect to 8 different attributes, including aspect ratio change, scale variation, partial occlusion, deformation, full occlusion, motion blur, viewpoint change, motion Blur, viewpoint change, and illumination variation. As shown in Figure 5, our tracker can better handle partial occlusion and deformation since multiple spatial attentions localize the most discriminative parts of the target. The learned part features

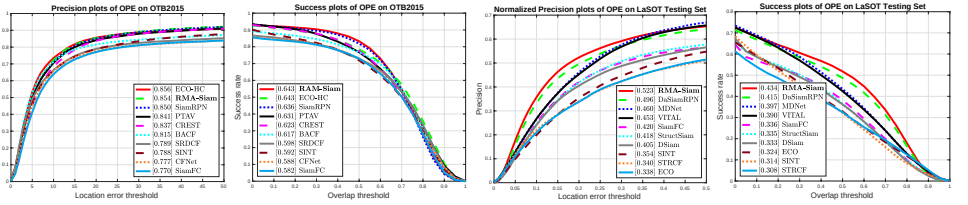

Figure 4: Evaluation results of different trackers on OTB-2015 and LaSOT.

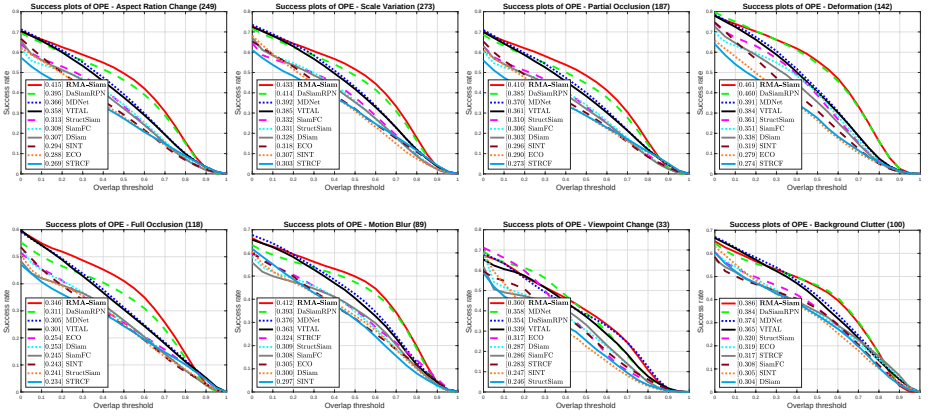

Figure 5: The success plots of eight attributes on LaSOT dataset.

are not influenced by the occluded part. Top performance on background clutter attribute further proves the effectiveness of the proposed semantic relation-aware compensation. It enhances the part features and makes them distinguishable from the background with similar local patterns. Our tracker also improves success plots of other attributes.

OTB-2015. The OTB-2015[45] provides a fair comparison on the accuracy and robustness with precision plots and success plots. We compare our trackers with 9 state-of-the-art trackers (ECO-HC[8], SiamRPN[22], PTAV[10], CREST[37], BACF[18], SRDCF[6], SINT[40], CFNet[41], and SiamFC[2]). The precision plots and success plots are show in Figure 4. Our tracker achieves success plots of 0.643.

TrackingNet. The TrackingNet [29] provides a large amount of data to assess trackers in the wild. We evaluate our trackers on testing dataset with 511 videos. Following [29], we use three metrics, including precision plots (PRE) , normalized precision plots (NPRE) and success plots (SUC), for evaluation. As shown in Table 3, our tracker achieves the best performance on PRE, NPRE, and SUC.

4.3 Ablation Analysis

We conduct experiments to verify our model designs. We use VOT benchmarks for the ablation analysis. As shown in Table 4, MA, AR, and RC denote the multiple attention, attention regularization, and relation-aware compensation respectively. We progressively add each module into the tracker. The preformance improvement of MA is very limited in EAO, due

Tracker	MDNet [30]	CFNet [41]	SiamFC [2]	ECO [8]	CSRDCF [5]	SAMF [25]	Staple [1]	BACF [18]	RMA-Siam
PRE	**0.565**	0.533	0.533	0.492	0.480	0.477	0.470	0.461	**0.594**
NPRE	**0.705**	0.654	0.663	0.618	0.622	0.598	0.603	0.580	**0.733**
SUC	**0.606**	0.578	0.571	0.554	0.534	0.504	0.529	0.523	**0.633**

Table 3: Comparisions on TrackingNet [29]. The best two results are highlighted in **red** and **blue**, respectively.

MA	AR	RC	VOT-2016			VOT-2017		
			A(↑)	R(↓)	EAO(↑)	A(↑)	R (↓)	EAO(↑)
			0.560	0.312	0.344	0.490	0.464	0.244
✓			0.608	0.308	0.350	0.582	0.482	0.256
✓	✓		0.611	0.275	0.357	0.587	0.454	0.269
✓		✓	0.624	0.252	0.375	0.585	0.431	0.279
✓	✓	✓	0.616	0.266	0.382	0.583	0.370	0.311

Table 4: Ablation study of our trackers on VOT benchmarks. MA, AR, and RC denote multiple attention, attention regularization, and relation-aware compensation.

to the fact that only part information will cause the tracker drift to background clutter with similar local patterns. Adding AR helps the tracker to detect diverse local patterns of the target and improves robustness compared with the second row and the third row. Moreover, when using MA and RC together, the performance further improves as shown in the fourth row. Finally, We conduct experiment with LC, AR and RC together. Compared with the baseline tracker SiamRPN, we obtain relative gains of 11.6% on VOT-2016 and 27.5% on VOT-2017 measured by EAO.

4.4 Qualitative results

Some qualitative results are demonstrated in Figure 6. Our tracker can better handle partial occlusion and deformation compared with the baseline tracker SiamPRN. When the target is partially occluded (*fish1*, *fernando*), our tracker can still track the target. Compared with the correlation filter based trackers in case of deformation (*fish1*, *motocross1*, *fernado*), our tracker can better estimate the size of target.

5 Conclusion

In this paper, we present relation-aware multiple attention module to boost the tracking performance. The RMA module discovers the most salient target parts with multiple attention maps, where the orthogonal regularization is applied to ensure the diversity of the local patterns. Relation-aware semantic compensation is then applied to capture the dependency among the part features. The resulting tracker benefits from the robust and discriminative part features. It achieves promising results on VOT series dataset, LaSOT, OTB-2015, and TrackingNet while maintains high tracking speed.

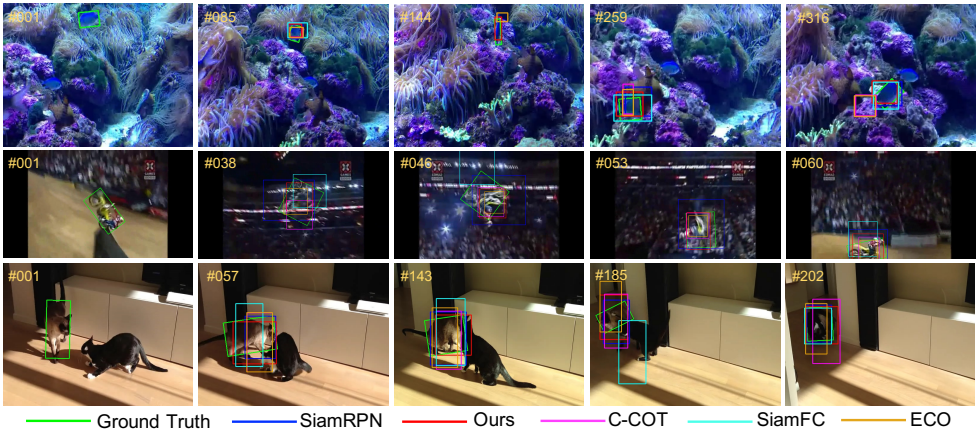

Figure 6: Visualization results. The videos in the first, second, third row are *fish1*, *motocross1*, and *fernando* in VOT-2017, respectively. The bounding boxes are not drawn if the tracker lost the target (except the first frame).

Acknowledgements

This work is partially supported by National Key R&D Program of China (No.2017YFA0700800), Natural Science Foundation of China (NSFC): 61876171 and 61572465, and Beijing Municipal Science and Technology Program: Z181100003918012.

References

[1] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip H. S. Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, 2016.

[2] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV Workshops*, 2016.

[3] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. A^2-nets: Double attention networks. In *NeurIPS*, 2018.

[4] Jongwon Choi, Hyung Jin Chang, Sangdoo Yun, Tobias Fischer, and Yiannis Demiris. Attentional correlation filter network for adaptive visual tracking. In *CVPR*, 2017.

[5] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, 2015.

[6] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Convolutional features for correlation filter based visual tracking. In *ICCV Workshops*, 2015.

[7] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016.

[8] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017.

[9] Dawei Du, Honggang Qi, Longyin Wen, Qi Tian, Qingming Huang, and Siwei Lyu. Geometric hypergraph learning for visual tracking. *arXiv: 1603.05930*, 2016.

[10] Heng Fan and Haibin Ling. Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In *ICCV*, 2017.

[11] Heng Fan and Jinhai Xiang. Robust visual tracking via local-global correlation filter. In *AAAI*, 2017.

[12] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. *arXiv:1809.07845*, 2018.

[13] Erhan Gundogdu and A. Aydin Alatan. Good features to correlate for visual tracking. *arXiv:1704.06326*, 2017.

[14] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *ICCV*, 2017.

[15] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. A twofold siamese network for real-time object tracking. In *CVPR*, 2018.

[16] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *ECCV*, 2016.

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[18] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, 2017.

[19] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, and Luka Cehovin Zajc. The visual object tracking vot2016 challenge results. In *ECCV Workshops*, 2015.

[20] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, and Luka Cehovin Zajc. The visual object tracking vot2017 challenge results. In *ICCV*, 2017.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[22] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.

[23] Feng Li, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming Hsuan Yang. Learning spatial-temporal regularized correlation filters for visual tracking. In *CVPR*, 2018.

[24] Shuang Li, Slawomir Bak, Peter Carr, and Xiaogang Wang. Diversity regularized spatiotemporal attention for video-based person re-identification. In *CVPR*, 2018.

[25] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV*, 2014.

[26] Yang Li, Jianke Zhu, and Steven C. H. Hoi. Reliable patch trackers: robust visual tracking by exploiting reliable patches. In *CVPR*, 2015.

[27] Si Liu, Tianzhu Zhang, Xiaochun Cao, and Changsheng Xu. Structural correlation filter for robust visual tracking. In *CVPR*, 2016.

[28] Ting Liu, Gang Wang, and Qinxiong Yang. Real-time part-based visual tracking via adaptive correlation filters. In *CVPR*, 2015.

[29] Matthias Müller, Adel Bibi, Silvio Giancola, Salman Al-Subaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018.

[30] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.

[31] Aaditya Prakash, James Storer, Dinei Florencio, and Chan Zhang. Repr: Improved training of convolutional filters. *arXiv:1811.07275*, 2018.

[32] Shi Pu, Yibing Song, Chao Ma, Honggang Zhang, and Ming-Hsuan Yang. Deep attentive tracking via reciprocative learning. In *NeurIPS*, 2018.

[33] Adam R. Kosiorek, Alex Bewley, and Ingmar Posner. Hierarchical attentive recurrent tracking. In *NeurIPS*, 2017.

[34] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *CVPR*, 2017.

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3): 211–252, 2015.

[37] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson W. H. Lau, and Ming-Hsuan Yang. Crest: Convolutional residual learning for visual tracking. In *ICCV*, 2017.

[38] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson WH Lau, and Ming-Hsuan Yang. Vital: Visual tracking via adversarial learning. In *CVPR*, 2018.

[39] Chong Sun, Huchuan Lu, and Ming-Hsuan Yang. Learning spatial-aware regressions for visual tracking. In *CVPR*, 2018.

[40] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *CVPR*, 2016.

[41] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip H. S. Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, 2017.

[42] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Steve Maybank. Learning attentions: Residual attentional siamese network for high performance online visual tracking. In *CVPR*, 2018.

[43] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[44] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018.

[45] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *TPAMI*, 37 (09):1834–1848, 2015.

[46] Rui Yao, Qinfeng Shi, Chunhua Shen, Yanning Zhang, and Anton Van Den Hengel. Part-based visual tracking with online latent structural learning. In *CVPR*, 2013.

[47] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv:1805.08318*, 2018.

[48] Yunhua Zhang, Lijun Wang, Jinqing Qi, Dong Wang, Mengyang Feng, and Huchuan Lu. Structured siamese network for real-time visual tracking. In *ECCV*, 2018.

[49] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, 2018.

[50] Zheng Zhu, Wei Wu, Wei Zhou, and Junjie Yan. End-to-end flow correlation tracking with spatial-temporal attention. In *CVPR*, 2018.