Discriminative Segmentation Tracking Using Dual Memory Banks

Fei Xie,1 Wankou Yang,1* Bo Liu,2 Kaihua Zhang,3 Wanli Xue4 Wangmeng Zuo5

1School of Automation, Southeast University
2JD Finance America Corporation
3Nanjing University of Information Science and Technology
4Tianjin University of Technology
5Harbin Institute of Technology

{jaffe03, wkyang}@seu.edu.cn, {kfliubo, zhkhua}@gmail.com, xuewanli@email.tjut.edu.cn, wmzuo@hit.edu.cn

Abstract

Existing template-based trackers usually localize the target in each frame with bounding box, thereby being limited in learning pixel-wise representation and handling complex and non-rigid transformation of the target. Further, existing segmentation tracking methods are still insufficient in modeling and exploiting dense correspondence of target pixels across frames. To overcome these limitations, this work presents a novel discriminative segmentation tracking architecture equipped with dual memory banks, i.e., appearance memory bank and spatial memory bank. In particular, the appearance memory bank utilizes spatial and temporal non-local similarity to propagate segmentation mask to the current frame, and we further treat discriminative correlation filter as spatial memory bank to store the mapping between feature map and spatial map. Without bells and whistles, our simple-yet-effective tracking architecture sets a new state-of-the-art on the VOT2016, VOT2018, VOT2019, GOT-10K and TrackingNet benchmarks, especially achieving the EAO of 0.535 and 0.506 respectively on VOT2016 and VOT2018. Moreover, our approach outperforms the leading segmentation tracker D3S on two video object segmentation benchmarks DAVIS16 and DAVIS17. The source code will be released at https://github.com/phiphiphi31/DMB.

1 Introduction

Visual tracking is a fundamental task in computer vision. In general, visual tracking aims at localizing the target in subsequent frames based the given bounding box in the first frame (Zhang and Peng 2020). So far, visual tracking still remains an challenging topic due to numerous factors such as deformation, occlusion and background clutter. Two dominant methodologies of deep trackers, Siamese correlation networks and discriminative correlation filters, mainly adopt a box-level representation for the target, making them limited in exploiting the fine-scale representation of the target. Moreover, bounding box representation is appropriate for axis-aligned transformation, but insufficient in handling complex and non-rigid transformation.

Several attempts have been made to develop segmentation-based trackers which leverage pixel-wise state estimation. SiamMask (Wang et al. 2019b) adds a segmentation branch to Siamese architecture for allowing joint learning of bounding box regression and object segmentation in training, but fails to combine them in the tracking stage. Later, D3S (Lukezic, Matas, and Kristan 2020) introduces a segmentation branch following VideoMatch (Hu, Huang, and Schwing 2018) and further combines online discriminative correlation filter (Bolme et al. 2010) (DCF) to fuse target classification and pixel-wise segmentation during inference. However, the segmentation branch in SiamMask is offline trained, while that in D3S is initialized using the first frame and then fixed during inference. Thus, both SiamMask and D3S fail in utilizing the temporal information to enhance the segmentation branch (See Fig. 2) and may lead to model under-fitting during tracking.

To overcome the limitations of SiamMask and D3S, this work presents to exploit dual memory banks, i.e., appearance memory bank and spatial memory bank, for improving tracking performance. While the DCF can be updated to cope with the appearance variation across frames for
filter-based deep trackers, little study has been given to consider temporal information in the segmentation branch during tracking. Obviously, the class-agnostic and one-shot segmentation branch adopted in SiamMask and D3S lack model flexibility to each video. In this paper, we present the appearance memory bank (AMB) for adapting the segmentation branch to temporal appearance variation while avoiding model drifting. In particular, we store keys and values of continuous frames in the AMB, and design a memory reader to compute the spatio-temporal attention to previous frames for each pixel in the query image (i.e., the current frame). Thus, albeit the network parameters of the memory module are fixed, we can dynamically update the memory bank to achieve better tradeoff between model generalization and flexibility (See Fig. 2). We further treat DCF as spatial memory bank (SMB) to model the mapping between feature map and spatial map. Moreover, the SMB helps to filter out the dirty samples in AMB while AMB provides SMB with more accurate target geometrical center. This mutual promotion on dual memory banks greatly boost the tracking performance. We also adopt box-to-segmentation training and testing strategy to mitigate inaccurate representation of bounding box initialization during tracking.

Experiments show that our tracker (DMB) sets a new state-of-the-art on the popular tracking benchmarks including VOT2016, VOT2018, VOT2019, GOT-10K and TrackerNet. As for video object segmentation (VOS), our approach also surpasses the leading discriminative segmentation tracker D3S on the DAVIS16 and DAVIS17 benchmarks. In comparison to template-based trackers, our approach can reduce the training data by more than an order of magnitude with improved tracking performance (See Fig. 1). As opposed to SiamMask and D3S, our approach is more effective on capturing and exploiting the temporal appearance changes of target. Comparisons can be viewed in Fig. 2.

The main contributions of this work are three-fold:

- We present appearance memory bank to capture and exploit the temporal appearance changes to enhance the segmentation branch of discriminative segmentation trackers.
- We treat the DCF as spatial memory bank and incorporate it with appearance memory bank to form our dual memory banks for bridging the gap between visual tracking and video object segmentation.
- Experiments show that our approach achieves the state-of-the-art results on five challenging tracking benchmarks and outperforms D3S and SiamMask on DAVIS16 and DAVIS17 for video object segmentation.

## 2 Related Work

### Template-based Deep Visual Tracking

The state-of-the-art template-based trackers generally can be grouped into two categories, i.e., Siamese correlation networks and DCF-based tracking approaches. Representative Siamese correlation trackers such as RPN-based (Ren et al. 2015) trackers (Li et al. 2018; ?) and anchor-free trackers (Xu et al. 2020; ?) usually consist of a classification branch for foreground-background estimation and a regression branch for box refinement. Recent filter-based trackers combine DCF (Bolme et al. 2010) with modified IoU/Net (Jiang et al. 2018). For example, ATOM (Danelljan et al. 2019) and DiMP (Bhat et al. 2019) utilize DCF for coarse localization, and exploit IoU prediction network for bounding box refinement. However, template-based trackers generally are training data-hungry to generalize well to previously unseen categories. Moreover, either the regression branch in Siamese trackers or the IoU prediction network in ATOM are offline trained and cannot adapt to temporal appearance changes during tracking. Furthermore, the bounding box representation of target restricts their ability in handling complex non-rigid transformation. In comparison, our approach uses the discriminative segmentation tracking framework to handling complex transformation and alleviate the data-hungry issue, and incorporate dual memory banks to cope with temporal appearance changes of target.

#### Discriminative Segmentation Tracking

Video object segmentation (VOS) methods usually are slow in speed and are not effective on handling the challenging factors in tracking scenarios, e.g., distractors and fast motion. It is mainly because the VOS task considers segmentation of large objects with limited appearance changes in short videos. SiamMask (Wang et al. 2019b) attempts to unify tracking and segmentation by adding a class-agnostic segmentation branch to detection-based tracker. D3S (Lukezic, Matas, and Kristan 2020) uses the DCF as the classification branch and a geometrically invariant template-based model for object segmentation. Our method also adopts the segmentation tracking architecture, and the dual memory banks are introduced to utilize temporal information.

#### Memory Network for Video Analysis

Recently, memory network has exhibited its merits in temporal modeling for video tasks. In visual tracking, MemTrack (Yang and Chan 2018) uses a dynamic memory network to adapt the template to the appearance variations. STM (Oh et al. 2019) applies a dynamic memory network to semi-supervised VOS and achieves appealing performance. MAST (Lai, Lu, and Xie 2020) exploits the memory module as spatio-temporal non-local attention in self-supervised video object segmentation. In this work, we present dual
memory banks to exploit the temporal appearance changes for bridging the gap between VOS and visual tracking.

3 Proposed Method
We present our discriminative segmentation tracking method with dual memory banks. In the following, we first introduce the overall pipeline of our approach, and then explain the appearance memory bank, spatial memory bank and decoder in details. Finally, we provide the dedicated designs in our tracker for visual tracking.

Overall Pipeline
Fig. 3 illustrates the pipeline of our approach. In our approach, each frame \( I_t \) is embedded into two triplets \( (Q_t, K_{A,t}, V_{A,t}) \) and \( (Q_t, K_{S,t}, V_{S,t}) \). In terms of memory network (Weston, Chopra, and Bordes 2015), \( Q, K, V \) refer to Query, Key, and Value, respectively. For the tracking and segmentation of current frame \( I_t \), an appearance memory encoder \( \text{Enc}^A_M \) is used to compute the appearance memory key and value representations \( \{ (K_{A,0}, V_{A,0}), ..., (K_{A,t-1}, V_{A,t-1}) \} \) for the previous frames \( \{ I_0, ..., I_{t-1} \} \). A spatial memory encoder \( \text{Enc}^S_M \) is introduced to extract the spatial memory keys \( \{ K_{S,0}, ..., K_{S,t-1} \} \). Following conventional tracking setting, the values \( \{ V_{S,0}, ..., V_{S,t-1} \} \) of spatial memory are computed based on the annotation of the first frame and the predicted target bounding boxes of the previous frames. Moreover, a query encoder \( \text{Enc}_Q \) is employed to obtain the query \( Q_t \) and the query value \( V_{Q,t} \) for the current frame \( I_t \).

Furthermore, the appearance memory reader module is adopted to generate the value \( V_{A,t} \) for the current frame. As for spatial memory, we treat DCF as a memory module, and use it to generate target location map. Subsequently, \( V_{A,t}, V_{Q,t}, \) target location map, and query encoder features are incorporated into a decoder to predict the segmentation mask of \( I_t \). Target bounding box can then be obtained from the segmentation mask, and the memory keys and values can also be updated and added to the appearance and spatial memory banks.

Appearance Memory Bank
Analogous to conventional memory network, our appearance memory bank consists of memory encoder \( \text{Enc}^A_M \), query encoder \( \text{Enc}_Q \), and memory reader. In particular, for each of the previous frames, the memory encoder takes the image and the frontal as well as the background segmentation masks as the input to produce the key and value. And the current frame is fed into query encoder to obtain query \( Q_t \) and query value \( V_{Q,t} \). Then, query \( Q_t \) is passed into memory reader to obtain the retrieved value \( V_{A,t} \) from appearance memory bank. Finally, \( V_{A,t} \) and \( V_{Q,t} \) are concatenated to form the read-out value \( R_t \). The memory encoder, query encoder and memory reader are explained as follows.

Memory Encoder. The input of memory encoder involves three components, i.e., an RGB frame, the foreground and background segmentation masks with probability between 0 and 1. Each component first goes through three convolutional layers individually and then be summed and fed into the backbone. Here we take ResNet-50 (He et al. 2016) as the backbone for both the memory encoder and the query encoder, and use the Conv4_4 layer as the common feature map \( f_{M} \) for computing the key and value. Then, the key and value can be obtained by respectively deploying their own
convolutional layer on the common feature map \(f_M\).

\[
K_A = \text{Key}_A(f_M), \quad V_A = \text{Val}_A(f_M).
\]  

(1)

During tracking, keys and values from all previous frames are stacked along the temporal order and are stored in the appearance memory bank.

**Query Encoder.** The query encoder \(\text{Enc}_Q\) takes the current frame \(I_t\) as the input to produce the query \(Q_t\) as well as the query value \(V_{Q,t}\). Analogous to memory encoder, we use the Conv4_e layer of ResNet-50 as the common feature map \(f_Q\). To generate the query \(Q_t\), a convolutional layer with linear activation is applied to reduce the number of channels to the 1/8 of \(f_Q\). The channel number of query value \(V_{Q,t}\) is a half of \(f_Q\).

\[
Q_t = \text{Que}_A(f_Q), \quad V_{Q,t} = \text{Val}_Q(f_Q).
\]  

(2)

**Memory Reader.** In the memory reader module, the keys and values \(\{(K_{A,0}, V_{A,0}), \ldots, (K_{A,t-1}, V_{A,t-1})\}\) of all previous frames, and the query and query value \((Q_t, V_{Q,t})\) of the current frame are used to produce the read-out value \(R_t\). In particular, the similarities between query \(Q_t\) and keys \(\{K_{A,0}, \ldots, K_{A,t-1}\}\) are utilized to measure the spatial and temporal non-local correspondence, which is then used to generate the retrieved value \(V_{A,t}\) for capturing temporal appearance changes. Then, the retrieved value \(V_{A,t}\) is computed based on the non-local attention mechanism formulated as follows,

\[
V_{A,t} = \sum_{t=1}^{t-1} \sum_{k=1}^{\frac{t}{2}} A_{t,j,k} V_{A,t},
\]  

(3)

\[
A_{t,j,k} = \frac{\exp \langle Q_t, K_{A,t-k} \rangle}{\sum_p \sum_{k=1}^{t-1} \exp \langle Q_t, K_{A,t-k-1} \rangle},
\]  

(4)

where \(i, j, p\) denote a spatial position of feature map, \(k\) denotes the index of a frame, and \(\langle \cdot, \cdot \rangle\) denotes the dot product between two vectors. We note that \(A_{t,j,k}\) measures the similarity between query and keys in spatio-temporal domain. Thus, albeit the network parameters of the appearance memory module are fixed, the memory bank can be updated during tracking, and the non-local spatio-temporal matching further makes it feasible for coping with temporal appearance variation of target. Furthermore, for enhancing the retrieved value \(V_{A,t}\), we concatenate it with the query value \(V_{Q,t}\) to obtain the read-out value,

\[
R_t = \text{concat}[V_{A,t}, V_{Q,t}],
\]  

(5)

where \(\text{concat}[\cdot, \cdot]\) denotes the concatenation operation.

In contrast to MAST (Lai, Lu, and Xie 2020) where the RGB image or segmentation mask are adopted as the value, we predict the appearance value as embedding feature map. And we encode both the query and the query value, where later is further concatenated with the retrieved value to get the read-out value.

**Spatial Memory Bank**

Inspired by (Danelljan et al. 2019), we introduce the spatial memory bank (SMB) to capture the temporal appearance changes of target for improving localization performance. In particular, the query encoder \(\text{Enc}_Q\) in AMB shares weights with the memory encoder and the query encoder for SMB. Let \(x_k = \text{Enc}_Q(I_k)\) be the feature map for a previous frame \(I_k\), and \(y_k\) be the corresponding spatial label. The DCF model can then be formulated as,

\[
f^* = \arg \min_f \sum_{k=0}^{t-1} \sum_{p} \|x_k^p, f - y_k\|^2 + \lambda \|f\|^2.
\]  

(6)
The feature map of the current frame $I_t$ is denoted by $x_t$. With the kernel tricks, we have,

$$R^t_{S,t} = \langle f^*, x^*_t \rangle = \sum_{k=0}^{t-1} \sum_{j} V^j_{S,k} A^{i,j,k}_{S,t},$$

where $i$, $j$, and $p$ denote a spatial position. We note that \{ $x_k | k = 0, ..., t-1$ \} and $x_t$ can be explained as the keys and query, while $V_{S,k}$ and $R_{S,t}$ are the value and read-out value in SMB. Thus, DCF can be explained as a special implementation of memory module to store the mapping between feature map and the read-out spatial map $R_{S,t}$. Moreover, the spatial map $R_{S,t}$ can serve as the spatial encoding of target localization, which is complementary to the read-out value in AMB. In our approach, we combine AMB and SMB to constitute the dual memory banks for improving segmentation and tracking performance.

**Decoder**

Fig. 5 illustrates the network architecture of the decoder. The read-out values of AMB and SMB are feature map and spatial map, which are then fed into the decoder module for predicting the final mask of the details frame. The spatial map serves as positional encoding for the feature map from AMB.

**Benefit from Dual Memory Banks**

In general, SMB is complementary to AMB and can collaborate to improve segmentation and tracking performance. We also present elaborate design to make the two banks cooperate well. Due to extreme challenging factors like blur and occlusion, we evaluate the state of the spatial map from SMB to filter the dirty samples out and store high quality samples in AMB. As shown in Fig. 6, the predicted mask from filtered samples is more accurate. On the contrary, the samples \{ $V_{S,k} | k = 0, ..., t-1$ \} stored in SMB are benefit from the geometric center generated by pixel-level representations. The localization of SMB can be more robust because of the geometrical robustness of target center.

4 Experiments

**Implementation details**

**Training phase.** For a better feature extraction ability, we firstly use image datasets instead of video sequences. ResNet-50 is initialized from the ImageNet pre-trained model. Similar to the training process in (Perazzi et al. 2017), we use image datasets annotated with object masks (Pascal VOC (Shetty 2016) and COCO (Lin et al. 2014)) to train our network. We apply image augmentations like random affine, flip and blur to the same image for generating a sequence of three images.

After training the encoder block, we use Youtube-VOS (Xu et al. 2018) and freeze the gradients of the encoder to train the decoder. We randomly sample 3 temporally ordered frames from the same video sequence and apply recurrent training strategy. The first frame and its mask are fed into memory encoder. The prediction of second frame is then stored in AMB for predicting the third frame. Then, the loss will be accumulated and backpropogated. We use randomly cropped 384384 patches for training and set the mini-batch to 4. We minimize the cross-entropy loss and mask IoU loss using Adam optimizer (Kingma and Ba 2014) with a fixed learning rate of 10⁻⁵. First-stage training process takes 120 epochs and decoder training takes 40 epochs using four NVIDIA TITAN XP GPUs. The total loss is shown below:

$$\mathcal{L} = \mathcal{L}_{BCE} (y^i_p, y^p) + \lambda \mathcal{L}_{IoU} (y^i_p, y^p),$$

where $\mathcal{L}_{BCE}$ indicates the binary cross-entropy loss, $\mathcal{L}_{IoU}$ indicates the standard IoU loss. $y^p$ in Eq. 9 is the predicted mask and $y^i$ is the mask label.

**Testing phase.** During inference, the sampling interval in appearance memory bank is set to 5. The output of our model is segmentation map and will be transferred to rotated box for tracking task.

**Box to segmentation strategy.** Segmentation-based tracker has to be robust towards bounding box initialization. By adding 1% bounding box label data called box-mask, we train our model to segment pseudo-mask with only one box-mask stored in memory banks. Then, the pseudo-segmentation mask will replace the box-mask cyclically in memory banks during tracking. This training and testing strategy greatly boost the performance in visual tracking.

**Evaluation**

**Evaluation on Tracking Datasets.** Our tracker was evaluated on five major short-term tracking benchmarks.
VOT-16

| Trackers | Update-Net | SPM | SiamMask-opt | SiamRPN++ | ATOM | DiMP-50 | Retina-MAML | D3S | Ocean-off | Ours(DMB) |
|----------|------------|-----|--------------|-------------|-----|--------|------------|-----|-----------|-----------|
|          | (2019)     |     | (2019a)      | (2019b)     | (2019) | (2019) | (2020)     | (2020) | (2020)   | (2020)   |
| A†       | 0.61       | 0.62 | 0.67         | 0.64        | 0.61  | -      | -          | 0.66 | -         | 0.684     |
| R†       | 0.21       | 0.21 | 0.23         | 0.20        | 0.180 | -      | -          | 0.131 | -         | 0.121     |
| EAO†     | 0.481      | 0.434 | 0.442       | 0.464      | 0.430 | -      | -          | 0.493 | -         | 0.535     |
|          |            |     |              |             |       |       |            |      |           |           |
| VOT-18   |            |     |              |             |       |       |            |      |           |           |
| A†       | -          | -   | 0.642        | 0.600       | 0.590 | 0.597  | 0.604      | 0.64  | 0.598     | 0.652     |
| R†       | -          | -   | 0.295        | 0.234       | 0.204 | 0.153  | 0.159      | 0.150 | 0.169     | 0.145     |
| EAO†     | 0.393      | -   | 0.387        | 0.414       | 0.401 | 0.440  | 0.452      | 0.489 | 0.467     | 0.506     |
|          |            |     |              |             |       |       |            |      |           |           |
| VOT-19   |            |     |              |             |       |       |            |      |           |           |
| A†       | -          | -   | 0.594        | 0.580       | 0.603 | -      | 0.570      | -    | 0.590     | 0.649     |
| R†       | -          | -   | 0.461        | 0.446       | 0.411 | -      | 0.366      | -    | 0.376     | 0.326     |
| EAO†     | 0.287      | -   | 0.292        | 0.329       | 0.301 | -      | 0.313      | -    | 0.327     | 0.356     |
|          |            |     |              |             |       |       |            |      |           |           |
| GOTE-10K |            |     |              |             |       |       |            |      |           |           |
| SR5†     | -          | -   | -            | 32.5        | 40.2  | 49.2   | -          | 46.2  | -         | 52.2      |
| AO†      | -          | -   | -            | 51.8        | 55.6  | 61.1   | -          | 59.7  | 59.2      | 61.5      |
|          |            |     |              |             |       |       |            |      |           |           |
| TrackingNet: |      |     |              |             |       |       |            |      |           |           |
| Prec.†   | 62.5       | -   | -            | 69.4        | 64.8  | 68.7   | -          | 66.4  | -         | 69.7      |
| SiamRPN++| 75.2       | -   | -            | 80.0        | 77.1  | 80.1   | 78.6       | 76.8  | -         | 79.4      |
| Succ.†   | 67.7       | 71.2 | -            | 73.3        | 70.3  | 74.0   | 69.8       | 72.8  | -         | 74.2      |

Table 1: Results on several benchmarks. Top-3 results of each dimension (row) are colored in red, blue and green, respectively.

Figure 7: Comparison of accuracy on VOT2018 for the following visual attributes: camera motion, illumination change, occlusion, size change and motion change. Empty means frames do not belong to any of the five attributes.

and compared with the state-of-the-art (SOTA) trackers: VOT2016 (Hadfield, Bowden, and Lebeda 2016), VOT2018 (Kristan et al. 2018), VOT2019 (Kristan et al. 2019), GOT-10k (Huang, Zhao, and Huang 2019), TrackingNet (Muller et al. 2018). Results are shown in Table 1.

VOT datasets are the most challenging and convincing evaluation tools in tracking. Each VOT dataset consists of 60 sequences. Performance is measured by accuracy (A), robustness (R) and the expected average overlap (EAO), which provides the overall performance metric.

**VOT2016**: Table 1 shows that our tracker outperforms all trackers on all three measures by a large margin. In terms of EAO, our tracker outperforms the strongest SOTA tracker D3S by 4.2 points and ATOM by 10.2 points. So far, DMB is the first one among published papers which breaks through the 0.53 in EAO and 0.68 in accuracy.

**VOT2018**: DMB outperforms all SOTA trackers in all three metrics. So far, our tracker is the first one among published papers which breaks through the 0.50 in EAO without redundant modules. Our tracker outperforms the D3S in EAO by 1.7 points, SiamMask in accuracy by 1.0 points and Ocean by over 2.4 points in robustness. As shown in Fig. 7, DMB is more accurate than other trackers towards challenging factors like occlusion, size and motion changes.

**VOT2019**: DMB is compared to the recent prevailing trackers. Our model surpasses all the competitive trackers in three metrics. Our tracker outperforms the most recent published Siamese correlation tracker Ocean by 2.9 points in EAO. The accuracy of our tracker outperforms the ATOM by 4.6 points. The results show that our tracking architecture has better performance towards both Siamese correlation trackers and filter-based trackers.

**GOT-10K**: GOT-10K is a recent large-scale dataset consisting of 10K video segments and 1.5 million axis-aligned bounding boxes. Average overlap (AO) and success rates at 75% threshold (SR75) are the main ranking metrics. As shown in Table 1, our tracker outperforms all competing trackers in two metrics and achieves the SOTA AO score of 61.5. Our tracker improves the SR75 by 3.0 points over the strongest filter-based tracker DiMP-50, while outperforming DiMP-50 by 0.4 points in terms of AO.

**TrackingNet**: TrackingNet dataset consists of 30K videos with more than 14 million dense bounding box annotations while the testing set contains 511 sequences. Three metrics are adopted: area under the success rate curve (AUC), precision (Prec.) and normalized precision (Prec.N ). DMB outperforms the strongest filter-based tracker DiMP-50 by 0.2 points in AUC while our accuracy surpasses the strongest segmentation-based tracker D3S by 3.3 points.

**Evaluation on VOS Datasets.** Our tracker is further evaluated on two popular VOS benchmarks DAVIS16 (Perazzi et al. 2016) and DAVIS17 (Pont-Tuset et al. 2017). Performance is evaluated by two metrics averaged over the sequences following official test protocol: mean Jaccard index \(J_M\) and mean F-measure \(F_M\). Our tracker is compared with the SOTA segmentation-based trackers and competitive VOS methods. From Table 2, our tracker outperforms the SOTA segmentation-based trackers D3S and SiamMask (Wang et al. 2019b) by a large margin. On the more challenging DAVIS17, our tracker even outperforms all the methods specialized to VOS task in mean of \(J_M\) and \(F_M\).
|   | J\text{I} | F\text{I} | J\&F\text{I} |
|---|---|---|---|
| Ours (DMB) | 79.0 | 75.5 | 77.3 |
| DS | 75.4 | 72.6 | 74.0 |
| SiamMask | 71.7 | 67.8 | 69.8 |
| OnAVOS (2017) | 86.1 | 84.9 | 85.5 |
| FAVOS (2018) | 82.4 | 79.5 | 80.9 |
| VM (2018) | 81.0 | - | - |
| OSVOS (2017) | 79.8 | 80.6 | 80.2 |
| OSMN (2018) | 74.0 | 72.9 | 73.5 |

Table 2: Comparison with segmentation-based trackers and VOS methods on DAVIS16 and DAVIS17.

Figure 8: DMB vs. SiamMask. Our tracker has better generalizaion ability towards unseen objects such as hand. Our tracker is more discriminative towards distractors and achieves a better performance on predicting object contour.

Ablation Study

To further show our contributions, we performed comprehensive ablation studies on VOT2018 and DAVIS16.

Temporal Information: In order to show the effectiveness of utilizing temporal information, we set different sampling interval of appearance memory bank. When sampling interval is 0, our tracker is the same as template-matching methods where only the first frame is used. As shown in Fig. 9, the all three measures drop by a large margin in comparison to the modes utilizing temporal information. No temporal information used causes 6.2 points performance drop in EAO, 8.6 points drop in Accuracy and 4.2 points drop in Robustness in contrast to storing every sample in memory bank. It further validates the superiority of our tracking architecture to the template-based trackers.

The amount of samples stored in memory bank also matters. When sampling interval is 1, our trackers reaches the top accuracy performance which is 0.663. Performance of EAO reaches the top which is 0.506 when sampling interval is 5 frames. Comparing to the 5 frames interval, 30 frames interval which is sparse reduces the EAO by 7.9 points. When the last frame always be added to appearance memory bank, our tracker boosts its overall performance EAO by 3.9 points and robustness performance by 2.8 points when sampling interval is 5 frames.

Box-to-Segmentation: As shown in Tab. 3, the box-to-segmentation training and testing strategy improves the EAO value by 1.4 point and the accuracy rises from 0.635 to 0.652. This validates that this strategy mitigates the inaccurate effect of bounding box initialization during tracking.

Positional Encoding: Inspired by CoordConv (Liu et al. 2018), we concatenate two coordinate channels to read-out features. Secondly, we simply do positional encoding as that in natural language processing. We add the single spatial matrix to the read-out features. By surpassing by 2 points in EAO, adding spatial matrix to the read-out is a more effective and simple way comparing to CoordConv.

Sample Filtering: As shown in Table 3, the collaboration between dual memory banks is significant to the overall performance. Without the samples filtered from SMB, the EAO drops from 0.506 to 0.421 when sampling interval equals to 5. The mean of J\&F on DAVIS16 also reduces from 77.3 to 69.1. This indicates that one single memory bank cannot handle these challenging tracking scenarios seperately.

5 Conclusion

In this work, we propose a new tracking architecture that fully exploits temporal information by considering visual tracking as video problems. Dual memory banks are successfully applied in both segmentation and localization sub-
tasks. Mutual promotion on two memory banks boosts the overall tracking performance and the box-to-segmentation strategy further bridges the gap between visual tracking and VOS. Without bells and whistles, our approach achieves state-of-the-art results on five challenging tracking benchmarks and outperforms the existing segmentation-based trackers in two VOS benchmarks.

References

Bhat, G.; Danelljan, M.; Gool, L. V.; and Timofte, R. 2019. Learning discriminative model prediction for tracking. In ICCV.

Bolme, D. S.; Beveridge, J. R.; Draper, B. A.; and Lui, Y. M. 2010. Visual object tracking using adaptive correlation filters. In 2010 IEEE computer society conference on computer vision and pattern recognition.

Caelles, S.; Maninis, K.-K.; Pont-Tuset, J.; Leal-Taixé, L.; Cremers, D.; and Van Gool, L. 2017. One-shot video object segmentation. In CVPR.

Cheng, J.; Tsai, Y.-H.; Hung, W.-C.; Wang, S.; and Yang, M.-H. 2018. Fast and accurate online video object segmentation via tracking parts. In CVPR.

Danelljan, M.; Bhat, G.; Khan, F. S.; and Felsberg, M. 2019. Atom: Accurate tracking by overlap maximization. In CVPR.

Hadfield, S.; Bowden, R.; and Lebeda, K. 2016. The visual object tracking vot2016 challenge results. Lecture Notes in Computer Science 9914:777–823.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In CVPR.

Hu, Y.-T.; Huang, J.-B.; and Schwing, A. G. 2018. Video-match: Matching based video object segmentation. In ECCV.

Huang, L.; Zhao, X.; and Huang, K. 2019. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; and Jiang, Y. 2018. Acquisition of localization confidence for accurate object detection. In ECCV.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pfugfelder, R.; Cehovin Zajc, L.; Vojir, T.; Bhat, G.; Lukezic, A.; Eldesokey, A.; et al. 2018. The sixth visual object tracking vot2018 challenge results. In ICCV.

Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Pfugfelder, R.; Kamarainen, J.-K.; Čehovin Zajc, L.; Drbohlav, O.; Lukezic, A.; Berg, A.; Eldesokey, A.; Kapyla, J.; and Fernandez, G. 2019. The seventh visual object tracking vot2019 challenge results.
Tracking by instance detection: A meta-learning approach. In CVPR.
Weston, J.; Chopra, S.; and Bordes, A. 2015. Memory networks. In ICLR.
Xu, N.; Yang, L.; Fan, Y.; Yang, J.; Yue, D.; Liang, Y.; Price, B.; Cohen, S.; and Huang, T. 2018. Youtube-vos: Sequence-to-sequence video object segmentation. In ECCV.
Xu, Y.; Wang, Z.; Li, X.; Yuan, Y.; and Yu, G. 2020. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In AAAI.
Yang, T., and Chan, A. B. 2018. Learning dynamic memory networks for object tracking. In ECCV.
Yang, L.; Wang, Y.; Xiong, X.; Yang, J.; and Katsaggelos, A. K. 2018. Efficient video object segmentation via network modulation. In CVPR.
Zhang, Z., and Peng, H. 2020. Ocean: Object-aware anchor-free tracking. In ECCV.
Zhang, L.; Gonzalez-Garcia, A.; Weijer, J. v. d.; Danelljan, M.; and Khan, F. S. 2019. Learning the model update for siamese trackers. In ICCV.
6 A. Qualitative analysis
Additional qualitative examples of tracking and segmentation are provided here. Video sequences are collected from the VOT2018, DAVIS16 and DAVIS17 datasets. Output of our tracker is segmentation mask in red color. A bounding box is generated from the predicted mask and shown in yellow. Our tracker predicts segmentation mask for DAVIS and rotated bounding box for VOT sequences. The following figures demonstrate corresponding characteristics of our tracker:

- Fig.10 shows video object segmentation results on DAVIS16 and DAVIS17 datasets.
- Fig.11 demonstrates that our tracker is robust to similar distractors and extreme complex background.
- Fig.12 demonstrates that our tracker can handle with the dim targets and appearance variations.
- Fig.13 shows our tracker’s generalization ability to track and segment unseen objects.
Figure 10: Video object segmentation on DAVIS16 or DAVIS17 datasets. Our tracker outperforms the leading segmentation-based trackers and predicts accurate masks under challenging scenes.

Figure 11: Examples of sequences with similar distractors and extreme complex background. It shows our tracker is robust to the similar objects even the distractors are closed to the targets. Our tracker can also handle with the complex background such as dark, raining and blur.
Figure 12: Examples of sequences with dim targets and appearance variations. As is shown in the sequence Bird, our tracker can predict accurate segmentation results even when target is extremely small. Moreover, our tracker achieves marvelous performance when fast-moving targets have extreme appearance deformation.

Figure 13: Examples of sequences with unseen targets. DNN-based trackers heavily rely on the pre-trained knowledge of targets. Our tracker utilizes the spatiotemporal information of every frame to enhance the generalization ability. From the examples of hand, blanket and glove, our tracker still tackles with those unseen targets well.