Learning to Track Objects from Unlabeled Videos —Supplementary Material—

Jilai Zheng[1] Chao Ma[1*] Houwen Peng[2] Xiaokang Yang[1]
[1] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2] Microsoft Research
{zhengjilai,chaoma,xkyang}@sjtu.edu.cn, houwen.peng@microsoft.com

In this supplementary material, we provide more details to complement the manuscript.

1. Box Sequence Generation

We propose to use dynamic programming (DP) to refine the candidate boxes $\mathcal{B}$ estimated from optical flow to produce a smooth box sequence $\mathcal{B}'$. The detailed algorithm for the DP process is presented in Alg. 1. At every step t in DP, the maximum path reward R_t is accumulated from the virtual starting box B_0 to box B_t, as detailed from line 3 to 6. The transition reward $R_{dp}(B_{t'}, B_t)$ is defined in Eqn. 3 of the manuscript. K indicates the constant reward for selecting a box.

Algorithm 1: DP-based Box Sequence Generation

Input: Candidate boxes $\mathcal{B} = \{B_t \mid 1 \leq t \leq L\}$
Output: Generated box sequence $\mathcal{B}' = \{B'_t \mid 1 \leq t \leq L\}$
1 $R_t \leftarrow K$ for $\forall 1 \leq t \leq L$; // Initialize the accumulated reward as K before DP
2 **for** $1 < t \leq L$ **do**
3 $t_{max} \leftarrow argmax_{1 \leq t' < t} \{R_{t'} + R_{dp}(B_{t'}, B_t)\}$; // Find optimal box transition at step t
4 $R_{max} \leftarrow R_{t_{max}} + R_{dp}(B_{t_{max}}, B_t) + K$; // Compute the accumulated reward until B_t
5 **if** $R_{max} \geq K$ **then**
6 $R_t \leftarrow R_{max}$;
7 Track back all candidate boxes on the path generated by DP with the highest accumulated reward ;
8 **for** $1 \leq t \leq L$ **do**
9 **if** B_t *is selected by DP* **then**
10 $B'_t \leftarrow B_t$; // Directly adopt candidate boxes selected by DP
11 **else**
12 Generate B'_t with linear interpolation ; // Smooth the generated box sequence

When generating box B'_t with linear interpolation, we choose two nearest candidate boxes B_u and B_v ($u < t < v$) selected by DP around B'_t as reference. Formally, the interpolation process is formulated as the following equation:

$$B'_t = \frac{v-t}{v-u} \cdot B_u + \frac{t-u}{v-u} \cdot B_v$$

2. Network Architecture

Following the conventional Siamese trackers [1, 2], the proposed naive Siamese tracker takes a $127 \times 127 \times 3$ template z_t and a $255 \times 255 \times 3$ search area x_t in frame I_t as inputs, and generates a $25 \times 25 \times 1$ classification score map $\mathcal{R}_{cls}$ and a $25 \times 25 \times 4$ regression map $\mathcal{R}_{reg}$ as outputs. For cycle memory training, N_{mem} memory search areas each sized $255 \times 255 \times 3$

* Corresponding author.

Figure A. Examples of training instances. The first two columns respectively denote the template and the search area for training the proposed naive Siamese tracker, while the remaining columns illustrate the memory search areas. Yellow boxes indicate the generated pseudo bounding boxes. Referring to row 1 - 4, our method succeeds in discovering foreground objects (animal, person, helicopter, etc.) under large appearance changes in complex scenes. Row 5 demonstrates a typical failure case, where the fast moving wave distracts the proposed method from locating the actual foreground object.

are inputted together with the Siamese pair. Based on the intermediate results of forward tracking, a memory queue sized $N_{mem} \times 7 \times 7 \times c$ is pooled from the deep features of N_{mem} memory search areas. Multi-scale correlation [3] between the deep features of x_t and the memory queue generates N_{mem} correlation maps denoted as $\{C^u_{corr} \mid 1 \leq u \leq N_{mem}\}$, altogether sized $N_{mem} \times 25 \times 25 \times c$. For each correlation map C^u_{corr}, we use two 3×3 convolution with padding of 1 to generate a confidence map C^u_{conf} and a value map C^u_{val}, both with the same size as C^u_{corr}. The finally integrated correlation map C, sized $25 \times 25 \times c$, is the weighted sum of $\{C^u_{val} \mid 1 \leq u \leq N_{mem}\}$, with $\left\{C^u_{conf} \mid 1 \leq u \leq N_{mem}\right\}$ being normalized as element-wise weights (see Eqn. 8). Note that softmax normalization is also conducted in an element-wise manner across all confidence maps. In other words, each softmax operation operates on N_{mem} elements collected from the same position of the N_{mem} confidence maps. C is finally converted into a $25 \times 25 \times 1$ memory map $\mathcal{R}_{mem}$ via convolution.

3. Example Training Instances

Fig. A provides real training instances. We can see that our box sampling method can discover the foreground objects under large appearance changes in the temporal span.

References

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

[2] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019.

[3] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020.