

DEEP FLOW COLLABORATIVE NETWORK FOR ONLINE VISUAL TRACKING

Peidong Liu¹ Xiyu Yan¹ Yong Jiang^{1,2,*} Shu-Tao Xia^{1,2}

¹Tsinghua Shenzhen International Graduate School, Tsinghua University
²PCL Research Center of Networks and Communications, Peng Cheng Laboratory
{lpd19, yanqy17}@mails.tsinghua.edu.cn; {jiangy, xiast}@sz.tsinghua.edu.cn

ABSTRACT

The deep learning-based visual tracking algorithms such as MDNet achieve high performance leveraging to the feature extraction ability of a deep neural network. However, the tracking efficiency of these trackers is not very high due to the slow feature extraction for each frame in a video. In this paper, we propose an effective tracking algorithm to alleviate the time-consuming problem. Specifically, we design a deep flow collaborative network, which executes the expensive feature network only on sparse keyframes and transfers the feature maps to other frames via optical flow. Moreover, we raise an effective adaptive keyframe scheduling mechanism to select the most appropriate keyframe. We evaluate the proposed approach on large-scale datasets: OTB2013 and OTB2015. The experiment results show that our algorithm achieves considerable speedup and high precision as well.

Index Terms— Deep learning, Visual tracking, Deep flow collaboration, Online learning, Keyframe scheduling

1. INTRODUCTION

Visual tracking task [1, 2] has attracted significant attention from researchers due to a wide range of potential applications such as VR, traffic control, robots, surveillance systems, etc. However, it is a challenging task because of the environmental variation, appearance variation of the target, and the high-efficiency requirements for some applications. An excellent tracker should consider both model robustness and effectiveness. For one thing, model robustness means the tracker performs well even when in complex environments such as background clutter, illumination variation, etc. Besides, the tracker can adapt to the appearance variation of the tracking object. For another thing, model effectiveness indicates the tracker achieves adequate speed for the applications.

With the development of the convolution neural network (CNN) [3, 4, 5, 6], deep learning-based methods are explored broadly in the visual tracking area. Benefitting from better feature representation extracted [7, 8, 9, 10] by CNN, deep learning-based trackers such as MDNet [11], HDT [12], and SINT [13] have obtained high accuracy in the large-scale benchmark, which means that they have excellent model robustness. However, most of these trackers haven't taken model effectiveness into full consideration. Specifically, they are time-consuming due to the complex architectures and large numbers of computations during feature extraction. Some previous works such as Real-time MDNet [14] and FlowTrack [15] have explored the model effectiveness. However, Real-time MDNet [14] realizes the model effectiveness at the cost of decreasing the model robustness. While FlowTrack [15] needs to spend time training the whole

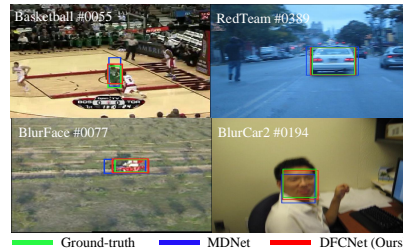


Fig. 1: Tracking results comparison of our approach DFCNet with MDNet [11] in the challenging scenarios.

model, including flow sub-network and feature sub-network, before the tracking process, which requires lots of computations.

In this work, we aim to propose an effective online algorithm that can fully consider model effectiveness to alleviate the time-consuming problem under the premise of maintaining model robustness at the same time. To be specific, we design a deep flow collaborative network (DFCNet) which utilizes inter-flow information in consecutive video sequences. As applying complex feature extractor to each frame is expensive, we speed up the tracking process by running the feature network only on sparse keyframes while other target states can be propagated through an optical flow map. Besides, we propose an effective keyframe scheduling mechanism to utilize appearance representation and temporal information. Fig. 1 shows DFCNet maintains robustness in complex scenes including background clutter, illumination variation, in-plane rotation, and out-of-plane rotation.

The contributions of this paper can be summarized as follows:

- We propose a deep flow collaborative tracking algorithm to alleviate the low-efficiency problem. Besides, an effective adaptive keyframe scheduling algorithm is developed, which can help propagate the flow map efficiently and fully utilize both flow information and appearance feature.
- In the experiments of OTB2013 [16] and OTB2015 [17], the proposed algorithm is about 60% faster than the baseline model MDNet [11] while maintains model robustness at the same time. Our tracker performs favorably against most of the existing popular trackers in terms of robustness.

2. PROPOSED METHOD

2.1. Network architecture

Our proposed model extends the tracker MDNet [11], champion of VOT2015 [18], to the sub-network for feature extraction. Yet, the

*: Corresponding author.

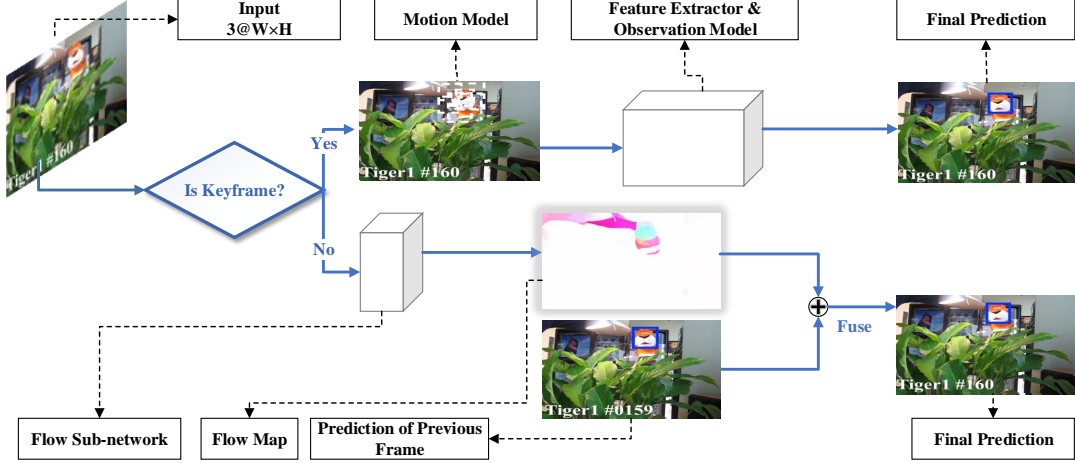


Fig. 2: The illustration of DFCNet Architecture, including the keyframe branch (motion model, feature extractor, and observation model) and non-keyframe branch (flow sub-network).

average speed of the MDNet [11] is around 1.55 fps [2] due to the generation of a large number of candidate samples and then the feature extraction through the deep neural network.

To speed up, DFCNet adopts a fast and effective method to avoid redundant computations and get accurate results by introducing optical flow. Only the sparse keyframes run the expensive feature network, and other target states are obtained through optical flow calculated with previous frames. As the difference between adjacent frames is limited, temporal information can be gained by optical flow. The specific network architecture is shown in Fig. 2. DFCNet first determines whether the current one is a keyframe. On the one hand, if it is exactly a keyframe, then we first generate a large number of candidate samples with the motion model, then extract features through the complex feature network, and finally obtain the estimated target with the observation model. On the other hand, if it is not a keyframe, we get the optical flow between the current frame and the previous frame through FlowNet2 [19] and then integrate the estimated result of the previous frame with the corresponding optical flow to determine the current target.

2.2. Adaptive keyframe scheduling mechanism

As for DFCNet, only keyframes pass through the feature extraction network. If all frames are determined to be keyframes, DFCNet degrades to the MDNet [11]. If only the first frame is a keyframe and the remaining are non-keyframes, due to the limited modeling of the appearance features, the tracker gets a poor accuracy. So it shows the significance of the keyframe selection strategy. Therefore, we propose a novel adaptive keyframe scheduling algorithm to get both speedup and high accuracy.

To begin with, DFCNet sets an interval K for scheduling mechanism to determine the fixed keyframes in a video sequence. For example, say $K = 3$, frames such as 1st, 4th, 7th and so on are considered as keyframes. The remaining frames are judged whether to be a keyframe or not in real-time based on the current tracking result. On the one hand, if the online tracking model scores higher than a given score threshold T , it indicates that the current tracking result is relatively accurate. Then the estimated optical flow is used to obtain the inter-frame information to complete the tracking process of the next frame. On the other hand, if the online tracking

Algorithm 1: Deep flow collaborative algorithm for on-line visual tracking.

Input: Initial target state B_0 , Number of frames in a video C , Number of point samples M , Score threshold T

Output: Estimated target state $\{B_i\}_{i=1}^N$

Initialize Score = $+\infty$.

for $i = 1$ to C **do**

if $i \bmod K == 1$ or $Score \leq T$ **then**

 /* Keyframe */

$D = N_{Feature}(B_{i-1})$.

 Calculate B_i according to Equation 1.

$Score = f^+(B_i)$.

else

 /* Non-keyframe */

$F_{i-1 \rightarrow i} = N_{Flow}(I_{i-1}, I_i)$.

 Sample M points $\{P_j\}_{j=1}^M$ from B_{i-1} .

 Propagate $\{P_j\}_{j=1}^M$ to $\{P'_j\}_{j=1}^M$ via flow field.

 Adjust $\{P'_j\}_{j=1}^M$ slightly to get B_i .

end

end

process has poor performance, the target of the next frame needs to be obtained through the feature network, which requires a lot more computations.

2.3. Online tracking algorithm

DFCNet is an online tracking algorithm. Online tracking refers to fine-tuning the tracker online with the addition of the sequences. In detail, DFCNet adopts the pre-trained tracking model in the beginning, and then the model is updated in the subsequent frames. Online tracking is suitable for the sequences as it can provide more accurate predictions with the increase of the input data.

DFCNet can be divided into several modules, including motion model, feature extractor, observation model, and model update. The online tracking algorithm is presented in Algorithm 1. The detailed tracking procedure is discussed in the following.

2.3.1. Motion model

Motion model adopts particle filter [20]. Based on the estimated target of the previous frame, the motion model generates Q candidate bounding boxes following Gaussian distribution, which may contain the target of the current frame.

In comparison, non-keyframes prevent running the motion model and generating large numbers of candidates.

2.3.2. Feature extractor and observation model

The feature extractor converts the raw RGB image into a semantic feature representation. It is the most critical part of a tracker as informative features can boost the tracking result significantly. Then the observation model judges whether the candidate is the target based on the features extracted.

In keyframes, DFCNet adopts a modified MDNet [11], to extract features by replacing the last multi-domain layer with a single-domain layer. The modified network is composed of three convolutional layers (Conv1-3) and three fully connected layers (FC4-6). During the online tracking process, Conv1-3 and FC4-5 layers use pre-trained parameters to initialize, and only three fully connected layers are updated. After the feature sub-network, $N_{Feature}$ evaluates Q candidate bounding boxes and obtains feature set D of those candidates, DFCNet chooses the one with the highest positive score to be the estimated target as equation 1 shows:

$$d^* = \operatorname{argmax}_d f^+(d), \quad (1)$$

where $d \in D$, f^+ is the positive score function, and d^* is the optimal candidate sample. While in the remaining non-keyframes, estimated targets are obtained by fusing the target state of the previous frame and optical flow, which is obtained from flow sub-network N_{Flow} . The overall procedure of DFCNet is presented in Algorithm 1. Further details are described below.

Shifting pixels in target state DFCNet estimates the target state of non-keyframes through optical flow obtained from FlowNet2 [19]. Specifically, DFCNet gets estimated flow from i^{th} to $i + 1^{th}$ frame $F_{i \rightarrow i+1}$ and uniformly samples M pixels in the flow map to propagate the target from i^{th} to $i + 1^{th}$ frame.

Bilinear interpolation in propagation As the coordinate of pixels propagated may be floating-point numbers, we apply the bilinear interpolation to get the optical flow values in the flow map. In particular, the pixels in the target state of i^{th} frame

$$\{(x_i^k, y_i^k) | k = 1, 2, 3, \dots, M\} \quad (2)$$

can propagate to $i + 1^{th}$ frame

$$\{(x_i^k + BIL(F_{i \rightarrow i+1}(x_i^k, y_i^k)))[x], \\ y_i^k + BIL(F_{i \rightarrow i+1}(x_i^k, y_i^k))[y] | k = 1, 2, 3, \dots, M\}, \quad (3)$$

where BIL represents bilinear interpolation.

Outliers removal and magnitude adjustment Propagation may bring in some outliers, which results in poor accuracy. DFCNet keeps only KR percent of propagated pixels to concentrate on the tracking object. As the appearance of objects in adjacent frames is similar, Hyperparameters adaptive ratio (AR) is used to balance the current and the previous target state to improve the robustness of the results.

2.3.3. Model update

The model updating strategy refers to that of MDNet [11], which mainly updates feature extractor. To fully consider model robustness and effectiveness, DFCNet applies long-term updates and short-term updates only on keyframes.

3. EXPERIMENTS

3.1. Settings

We evaluated DFCNet on OTB2013 [16] and OTB2015 [17]. The feature sub-network is pre-trained on VOT2015 [18], which excludes video sequences in OTB2015 [17]. The flow sub-network is pre-trained for video recognition. In online tracking, adaptive keyframe interval K and score threshold T is set to 3 and 10, numbers of candidate samples Q and pixels in bounding box M is set to 256 and 100, the ratio of reserved pixels KR and the adaptive ratio AR is set to 0.9 and 0.4, respectively. The one-pass evaluation (OPE) is applied to compare DFCNet with other trackers. For a fair comparison, all the tracking results use the reported results. Our algorithm is implemented in Pytorch and runs at a PC with 2.2GHz CPU and GTX1080 GPU.

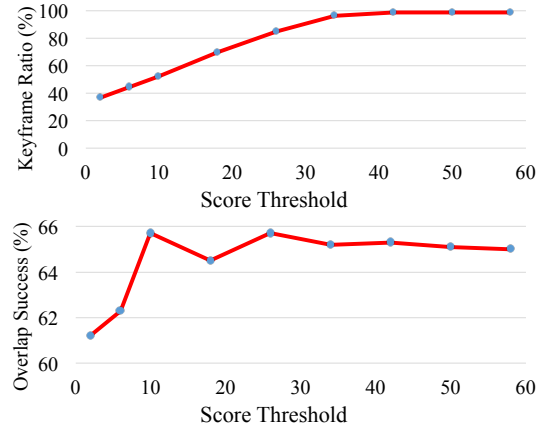


Fig. 3: The above and below illustrations indicate the relationship between keyframe ratio and score threshold, overlap success and score threshold, respectively.

3.2. Ablation studies

In this experiment, ablation studies are employed to illustrate the effectiveness of the adaptive keyframe scheduling algorithm. Fig. 3 illustrates how the keyframe ratio and overlap success correlate with the score threshold on OTB2015. The above illustration represents when the score threshold gets higher, the ratio of keyframes increases. It can be explained that the score threshold affects the number of keyframes directly. The below illustration shows when the score threshold gets higher, the overlap success ratio first increases, and then converges to that of MDNet [11] in the end. It can be appropriately interpreted that there exists an extreme value for the score threshold in our framework. In other words, DFCNet achieves both model effectiveness and robustness with a proper keyframe ratio. When the percentage of keyframes is low (lower than the proper ratio), DFCNet does not have adequate appearance information, which results in poor performance. However, when

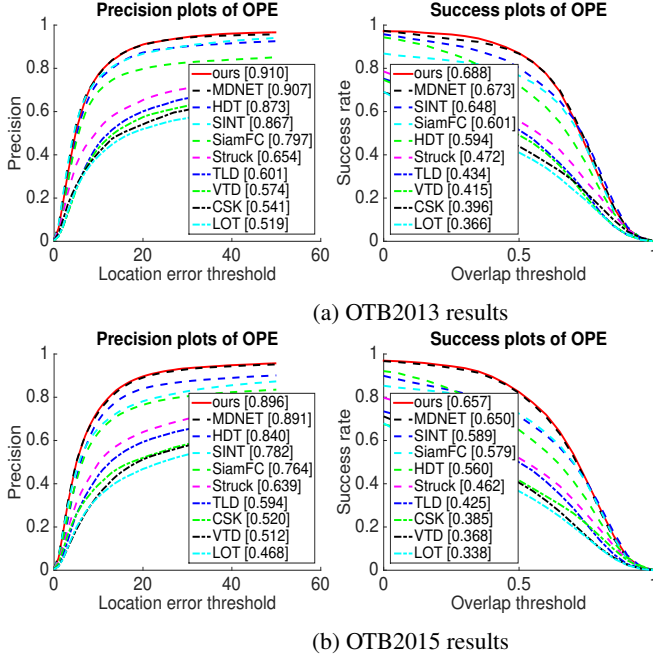


Fig. 4: Precision and success plot on OTB. The numbers in the legend indicate the representative precision at 20 pixels for precision plots, and the area-under-curve scores for success plots.

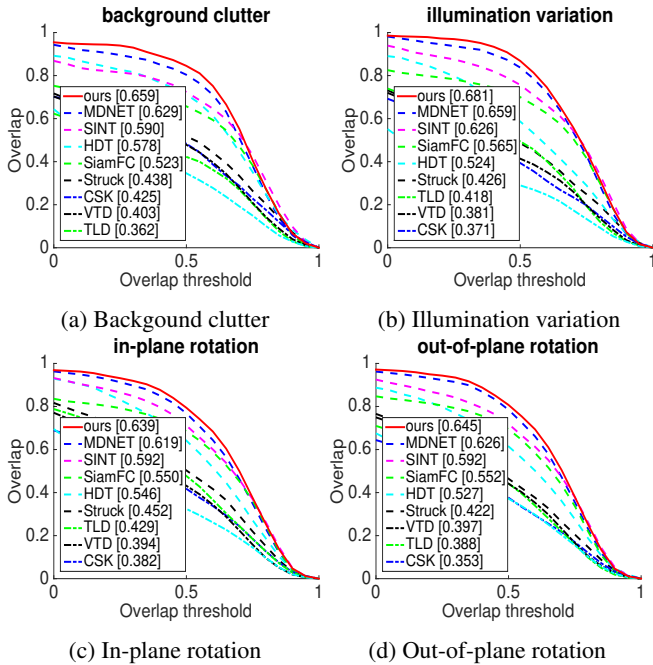


Fig. 5: The success plots with specific challenge attributes.

the percentage maintains around the proper ratio, DFCNet obtains both sufficient inter-frame information and object appearance features, which leads to high accuracy. At the same time, as the flow computation is less expensive than feature extraction, the model is substantially faster than the baseline model MDNet [11].

Table 1 records the tracking accuracy and speed of the DFC-

Net with the variation of the score threshold on OTB2015. DFCNet/w indicates a method without an adaptive keyframe mechanism. DFCNet-**number** represents a standard DFCNet with an adaptive keyframe mechanism, and the score threshold is set to **number**. It shows the effectiveness of the adaptive keyframe mechanism.

KFR = KeyFrame Ratio, OS = Overlap Success, SU = SpeedUp

Metric	KFR (%)	OS (%)	Speed (fps)	SU (%)
DFCNet/w	33.3	59.0	4.06	120.9
DFCNet-2	36.8	61.2	3.77	104.1
DFCNet-6	44.5	62.3	3.53	91.8
DFCNet-10	52.4	65.7	2.95	60.3
DFCNet-18	69.7	64.5	2.50	36.1
DFCNet-26	85.0	65.7	2.21	20.2
DFCNet-34	96.3	65.2	1.90	3.4
DFCNet-42	99.3	65.3	1.86	1.0
DFCNet-50	99.9	65.1	1.84	0
DFCNet-58	100	65.0	1.84	0
MDNet	-	65.0	1.84	-

Table 1: The table of overlap success and speed with the variation of score threshold on OTB2015 (DFCNet vs. MDNet). The line with bold is the setting we used in the examination on OTB.

3.3. Results on OTB

OTB2013 [16] has 50 annotated videos and OTB2015 [17] extends to 100. In this experiment, we compare our method against trackers that published at top conferences and journals, including MDNet [11], HDT [12], SINT+ [13], SiamFC [21], Struck [22], TLD [23], CSK [24], LOT [25], VDT [26].

Fig. 4 illustrates the overlap success and precision plots. It clearly shows that DFCNet slightly exceeds MDNet and outperforms the popular trackers. For further performance analyses, we also represent the results on various challenge attributes in OTB2015, such as background clutter, illumination variation, in-plane rotation, and out-of-plane rotation. Fig. 5 shows that our tracker effectively handles these challenges while others obtain relatively low scores.

4. CONCLUSION

In this work, we propose a flow collaborative network that only runs the complex feature network on sparse keyframes and propagates the features to other frames via flow map. Besides, an adaptive keyframe scheduling mechanism is employed to maximize the benefits of both appearance and temporal information. The approach is validated on benchmarks OTB2013 [16] and OTB2015 [17]. It is around 60% faster than MDNet [11] on OTB2015, which indicates the effectiveness of our method. Moreover, DFCNet performs favorably against existing popular trackers in accuracy and significantly advances the practice of visual tracking tasks.

5. ACKNOWLEDGMENTS

This work is supported in part by the National Key Research and Development Program of China under Grant 2018YFB1800204, the National Natural Science Foundation of China under Grant 61771273, the R&D Program of Shenzhen under Grant JCYJ201805 08152204044, and the research fund of PCL Future Regional Network Facilities for Large-scale Experiments and Applications (PCL2018KP001).

6. REFERENCES

- [1] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2013.
- [2] Peixia Li, Dong Wang, Lijun Wang, and Huchuan Lu, "Deep visual tracking: Review and experimental comparison," *Pattern Recognition*, vol. 76, pp. 323–338, 2018.
- [3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012, pp. 1097–1105.
- [4] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.
- [5] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
- [7] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014, pp. 818–833.
- [8] Samuel Ritter, David GT Barrett, Adam Santoro, and Matt M Botvinick, "Cognitive psychology for deep neural networks: A shape bias case study," in *ICML*, 2017, pp. 2940–2949.
- [9] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," in *ICLR*, 2019.
- [10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge, "Texture and art with deep neural networks," *Current Opinion in Neurobiology*, vol. 46, pp. 178–186, 2017.
- [11] Hyeonseob Nam and Bohyung Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*, 2016, pp. 4293–4302.
- [12] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, Jongwoo Lim, and Ming-Hsuan Yang, "Hedged deep tracking," in *CVPR*, 2016, pp. 4303–4311.
- [13] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders, "Siamese instance search for tracking," in *CVPR*, 2016, pp. 1420–1429.
- [14] Ilchae Jung, Jeany Son, Mooyeol Baek, and Bohyung Han, "Real-time mdnet," in *ECCV*, 2018, pp. 83–98.
- [15] Zheng Zhu, Wei Wu, Wei Zou, and Junjie Yan, "End-to-end flow correlation tracking with spatial-temporal attention," in *CVPR*, 2018, pp. 548–557.
- [16] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *CVPR*, 2013, pp. 2411–2418.
- [17] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.
- [18] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felberg, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, Gustavo Hager, Georg Nebehay, and Roman Pflugfelder, "The visual object tracking vot2015 challenge results," in *ICCVW*, 2015, pp. 1–23.
- [19] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *CVPR*, 2017, pp. 2462–2470.
- [20] Petar M Djuric, Jayesh H Kotecha, Jianqui Zhang, Yufei Huang, Tadesse Ghirmai, Mónica F Bugallo, and Joaquin Miguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003.
- [21] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr, "Fully-convolutional siamese networks for object tracking," in *ECCV*, 2016, pp. 850–865.
- [22] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L Hicks, and Philip HS Torr, "Struck: Structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096–2109, 2015.
- [23] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2011.
- [24] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *ECCV*, 2012, pp. 702–715.
- [25] Shaul Oron, Aharon Bar-Hillel, Dan Levi, and Shai Avidan, "Locally orderless tracking," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 213–228, 2015.
- [26] Junseok Kwon and Kyoung Mu Lee, "Visual tracking decomposition," in *CVPR*, 2010, pp. 1269–1276.