Multi-Scale Recurrent Tracking via Pyramid Recurrent Network and Optical Flow

Ding Ma[1]
madingcs@hit.edu.cn
Wei Bu[2]
buwei@hit.edu.cn
Xiangqian Wu[1]
xqwu@hit.edu.cn

[1] School of Computer Science and Technology
Harbin Institute of Technology
Harbin, China

[2] Department of New Media Technologies and Arts
Harbin Institute of Technology
Harbin, China

Abstract

The target in a tracking sequence can be considered as a set of spatiotemporal data with various locations in different frames, and the problem how to extract spatiotemporal information of the target effectively has drawn increasing interest recently. In this paper, we exploit spatiotemporal information by different-scale-context aggregation through the proposed pyramid multi-directional recurrent network (*PRNet*) together with the *FlowNet*. The *PRNet* is proposed to memorize the multi-scale spatiotemporal information of self-structure of the target. The *FlowNet* is employed to capture motion information for discriminating targets from the background. And the two networks form the *FPRNet*, being trained jointly to learn more useful spatiotemporal representations for visual tracking. The proposed tracker is evaluated on OTB50, OTB100 and TC128 benchmarks, and the experimental results show that the proposed *FPRNet* can effectively address different challenging cases and achieve better performance than the state-of-the-art trackers.

1 Introduction

Visual object tracking plays a fundamental role in computer vision. And visual tracking has been widely applied in various fields such as autonomous driving, human-computer interaction and video surveillance, etc. The core task for a single object tracking is to track an arbitrary target with a bounding box in constantly changing sequences being influenced by some factors including scale variations, complex backgrounds and occlusions (see Figure 1).

Inspired by the great success of convolutional neural networks (CNNs), many CNN-based trackers [4, 22, 25, 33, 36, 38] are proposed for the powerful feature representation. Here, some trackers [33, 38] achieve high performance by a feed-forward pass through the CNNs pretrained on a large-scale dataset building block for image classification. However, the fundamental gap between visual tracking and image classification will make it suboptimal in capturing suitable representations for tracking tasks to some extent. Besides, the speed of such trackers is approximately 1 fps though the pre-trained network is without a

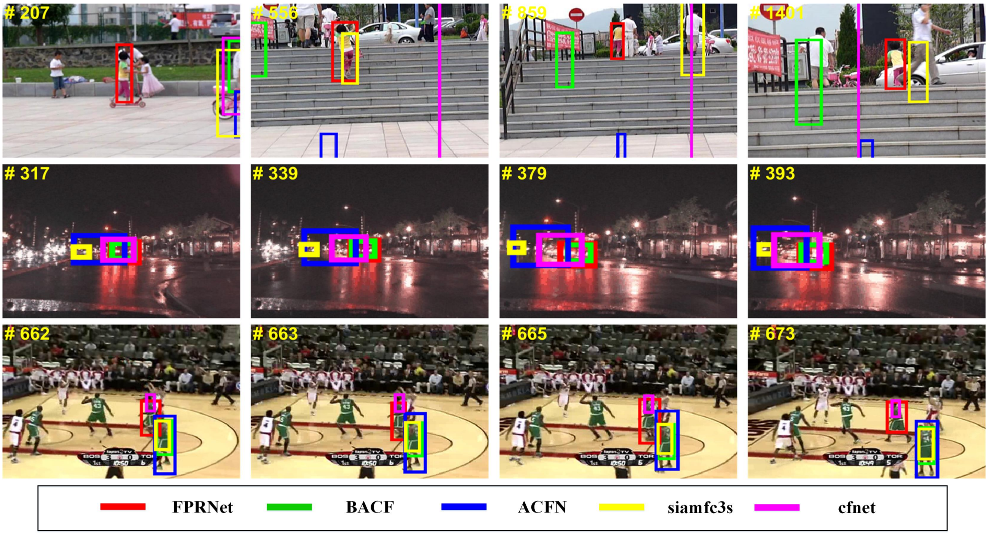


Figure 1: A comparison of the proposed *FPRNet* with the state-of-the-art trackers BACF [17], ACFN [7], siamfc3s [4] and cfnet [44] with interference factors, scale variations, complex backgrounds and occlusions. The shown sequences (*Girl2*, *CarDark*, *Basketball*) are from OTB benchmark. And our algorithm efficiently handles these challenging situations compared to existing approaches.

backward pass operation. Recently, the end-to-end learning convolutional architecture has gained growing attention to cope with the strict constraints on the speed and precision in tracking. [4, 25] significantly improve the speed of tracking process by designing a fully offline convolutional network without considering the model update. However, such real-time trackers are not taking full consideration of the temporal and spatial relation information. In order to improve the quality of [4, 25], [48] proposes a recurrent filter for memorizing the various variations of the target appearance during the tracking process.

It is observed that among trackers mentioned above, the inputs of frame t are pairwise image patches. One is the image patch of the tracked target in frame $t-1$, and the other one is the searching image patch in frame t. And the search image patch takes the local spatial relation information which depend on the result of frame $t-1$. We argue that different-scale-context in the whole image provides different spatial relation clues and should be fully utilized in a proper manner for visual tracking.

To incorporate comprehensive context clues, we propose a *pyramid recurrent network (PRNet)*. In each scale of context, as many as 8-directional RNN are proposed to gather the contextual information of the target. Each directional RNN can extract contextual dependencies between parts of the target, which makes it more discriminative to not only background objects but also similar distractors. Besides, by considering that the different layers in the CNNs capture different image perspectives, the proposed multiple pyramid RNNs are deployed in different layers of CNNs, which compose into a biger pyramid RNN to strengthens robustness of the proposed model. Furthermore, the pyramid RNN architecture aggregates the local and global spatial relation information, which make the final prediction more reliable.

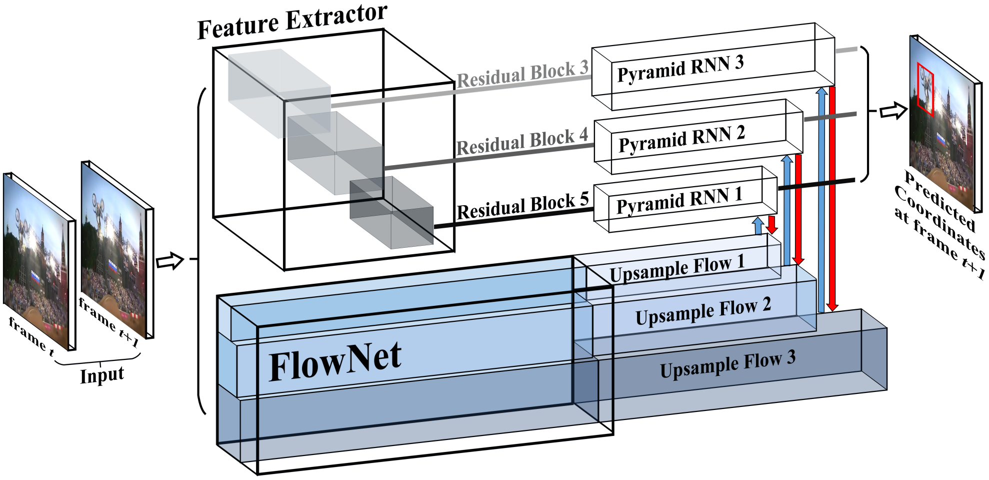


Figure 2: An overview of the proposed *FPRNet*. The *FPRNet* is composed of 2 subnetworks including the proposed *PRNet* and the *FlowNet*. The *PRNet* consists of the feature extraction part (*ResNet101*) and 3-different-scale pyramid RNNs which are derived from the Residual Block 3, Residual Block 4 and Residual Block 5, respectively. The different-scale feature maps from the 2 subnetworks are concatenated to train the unified *FPRNet* (resize the feature maps to the same size). The figure is best viewed in color.

Motion and inter-frame motion in *optical flow* can enhance the tracking performance during challenges such as occlusion and deformation. Although some trackers utilize *optical flow* to improve tracking performance [18], the flow feature is lack of self-perspectives within the target (which is characterized in *PRNet*). Besides, the optical flow itself is a challenging problem and is often inaccurate, and thus the provided information does not always provide precise motion information for tracking tasks. In this paper, a unified network *FPRNet* is composed of the *FlowNet* and the *pyramid recurrent network (PRNet)*, which learns more suitable representations for tracking. The overview of the proposed tracker is shown in Figure 2.

To evaluate our proposed *FPRNet*, we carry out extensive experiments on OTB50, OTB100 and TC128 benchmarks. On the challenging OTB50, OTB100 and TC128, the proposed tracker achieves the best performance in area under curve (AUC).

The main contributions of this work are summarized as follows:

I, We propose a pyramid multi-directional recurrent network (*PRNet*) to memorize the different-scale-context based on information of self-structure within the object.

II, We propose an end-to-end flow pyramid recurrent framework *FPRNet* to improve the spatiotemporal representations for tracking.

III, Quantitative and qualitative evaluation demonstrate the outstanding performance of our tracking algorithm compared to the state-of-the-art techniques in 2 public benchmarks: OTB50 [45], OTB100 [46] and TC128 [31].

The rest of the paper is organized as follows. We briefly review related work in Section 2. The detailed configuration of the proposed algorithm is described in Section 3. The training details are shown in Section 4. Section 5 illustrates experimental results on the 2 large tracking benchmarks. Finally, conclusions are drawn in Section 6.

2 Related Work

2.1 Visual Object Tracking

Depending on the appearance model, the tracking methods can be divided into generative and discriminative model. In one way, generative model searches the most similar candidate of the target with minimal reconstruction error including [21, 35, 39], etc. In another way, discriminative model formulates a binary classification problem to separate the target from background. Quite a few discriminative model based trackers have been proposed, such as [1, 19, 23].

In order to overcome the high computational burden and run in real-time, a series of correlation filter based trackers have been proposed [6, 8, 9, 26, 27]. The [6] proposes a correlation filter tracker via learning the minimum output sum of squared error (MOSSE). Henriques *et al.* [26] introduce a well-known kernelized correlation filters (KCF) tracker. And DSST [9] is proposed to solve the scale problem. MUSTer [27] strengthens the model stability by a short-long term strategy. C-COT [13], ECO [11] and DeepSRDCF [10] employ deep features to improve the robustness of the models.

2.2 CNN-based Tracking

Benefiting from the powerful representations of the deep networks, the CNNs have been recently introduced in visual tracking. Bohyung Han *et al.* [36] take full advantage of the end-to-end learning, and achieve the state-of-the-art performance by online updating the network. The TCNN [37] is proposed to strengthen the stability by managing multiple target appearance models in a tree structure. In order to improve the inter-class classification ability, Heng Fan *et al.* [15] adds the recurrent neural networks (RNNs) into CNNs to aware the self-structure of the object. In addition, more robust appearance model is introduced in [22]. Such trackers often meet the high computational burden and difficult to be implemented in real-time. On the contrary, [4, 20, 25, 43, 44] have been proposed to accelerate the speed of tracking process. [43] adopts a two-stream architecture for tracking. [25] proposed a two-stream tracker called GOTURN, which is trained to regress directly from dual frames to the location in the current frame of the target in the query sample. [4] introduced a fully-convolutional Siamese network (two stream architecture), which maps an exemplar of the target and a larger search area of the second frame to a response map. Different from these trackers, our *pyramid recurrent network (PRNet)* utilizes 8-directional RNN to gather the contextual information of the target in each level of the pyramid. The different-scale-contextual dependencies between parts of the target improves discriminative ability to not only background objects but also similar distractors.

2.3 Optical Flow for Vision Tasks

Many vision tasks [18, 30, 42, 49, 51] have been proposed with the help of *optical flow* for extracting motion information. In video detection, [51] presents flow-guided feature aggregation, an end-to-end learning framework for video object detection. In video action recognition, [42] (OFF) introduces a compact motion representation for video action recognition

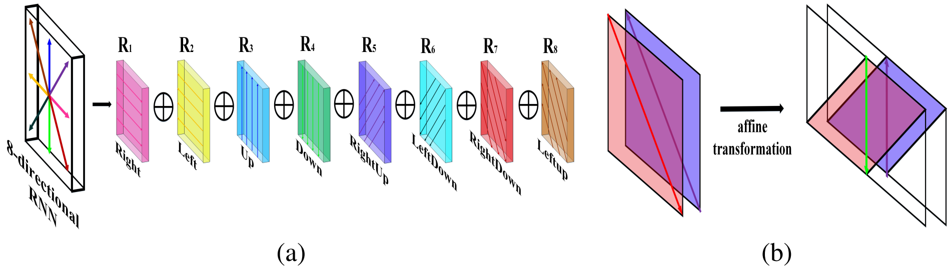


Figure 3: (a) An overview of the proposed 8-directional RNN architecture. Each directional RNN are concatenated to form the final 8-directional RNN, and ⊕ is a concatenation operator; (b) For the *RightDown/RightUp*-directional RNN, we adopt the affine transformation on feature maps, and set the same operations on *Down/Up*-directional RNN

.

which enables the network to distill temporal information through a fast and robust approach. In video saliency detection, [30] enhances the temporal coherence of the per-frame feature by exploiting both motion information in terms of *optical flow* and sequential feature evolution encoding in terms of *LSTM* networks. In video segmentation, [49] formulates a unified segmentation framework based on a compositional model by combining saliency flow detection with motion estimation. And in tracking task, [18] just employs flow feature which is not trained end-to-end. By contrast, we apply the *FlowNet* [14] to characterize the motion information of the target, and the unified end-to-end tracking framework *FPRNet* is composed of *PRNet* and the *FlowNet*, which learns more suitable spatiotemporal representations for tracking.

3 The Proposed *FPRNet*

In our observation, some tracking failures are partially related to contextual relationships within the different receptive fields. Thus a deep network with a compresive different-scale-context clues can much improve the discriminative ability for separating the target from complex background and similar distractors. Besides, the motion information from the *FlowNet* is utilized to enhance the spatiotemporal representations. In the rest of this section, we will introduce the details of 8-directional RNN architecture in Section 3.1. Then, the proposed *pyramid recurrent network (PRNet)* will be shown in Section 3.2. Finally, the unified *pyramid recurrent network (FPRNet)* will be illustrated in Section 3.3.

3.1 8-directional RNN

The details of our proposed 8-directional RNN are shown in Figure 3. For our problem, we extend the traditional RNN to two dimensions by computing along each row and each column of the image with the RNN. And the IRNN [29] is selected for its fast speed and small parameter search space.

In our 8-directional IRNN, the feature map is represented by a graph. The vertex in each graph relies on its predecessor. Based on the local input, the hidden output acts as a nonlinear function on its predecessor. Each IRNN shares the same structure. The IRNNs in 8

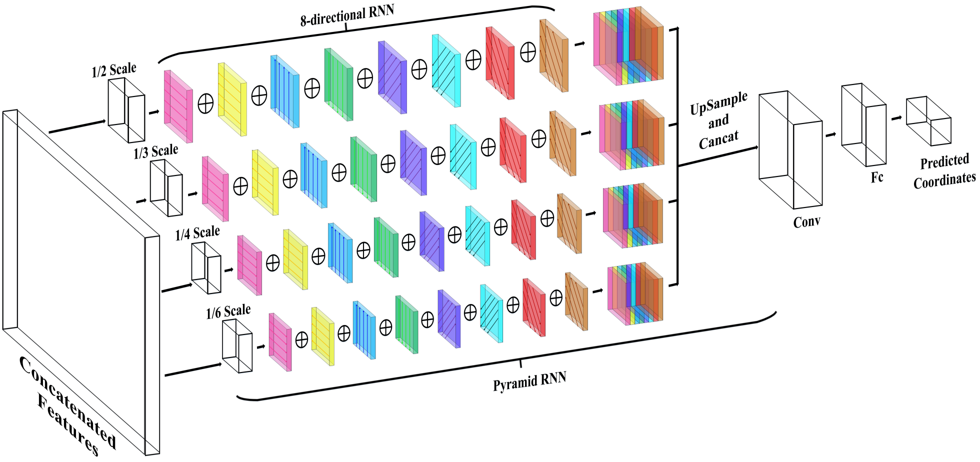


Figure 4: The example of pyramid RNN architecture which is derived from the Residual Block.

directions are represented as follows:

$$h^*_{i,j} \leftarrow \max(\mathbf{W}^*_{hh} h^*_{i,j-1} + h^*_{i,j}, 0) \tag{1}$$

where $*$ denotes the direction of RNN. For each direction, we gather all IRNNs together with a single matrix multiplication. As is shown in Figure 3a, R_1, R_2, R_3, R_4, R_5, R_6, R_7 and R_8 represent the RNN in 8 different directions, respectively: right, left, up, down, rightup, leftdown, rightdown and leftup. After the 8-directional RNN, each cell on the output layer is the combination of local and global, and can be as the global summary of the object feature map. And the output layer $\mathbf{o}$ can be expressed as follows:

$$\mathbf{o} = g(\sum_{R_1,\ldots,R_8} V h^*_{i,j} + c) \tag{2}$$

where V is the matrix parameters, c is the bias term, and $g(\cdot)$ is the elementwise nonlinear activation functions.

3.2 *PRNet*

On the contrary to [2, 47], we add another 4-directional IRNN (rightup, leftdown, rightdown and leftup) to gather more directions of IRNNs for coping with the various appearance variation of the target (see Figure 3b). For the RightDown IRNN given a 7×7 feature map, firstly, we do affine transformation on the feature map. Then, the size of the feature map has changed from 7×7 to 7×13. The white area of the affine transformed feature map is filled with 0, and the other areas are the translation shift. As a result, the RightDown IRNN operation of the 7×7 feature map turns into a down IRNN operation of the 7×13 feature map.

Finally, a inverse transformation is performed to restore the size of the affine transformed feature map to 7 × 7. Similarly, IRNN in the other three directions takes the same operation.

The example of *pyramid RNN architecture* which is derived from the Residual Block is shown in Figure 4. Given a residual block, we first gather the feature maps of a residual block and upsampled flow, then the concatenated features are fed into a pyramid recurrent architecture to harvest different sub-region recurrent representations. In the *pyramid RNN architecture*, each pyramid level separates the feature map into different sub-regions and multi-directional recurrent representations of the feature map is made in each sub-region. The output of different levels in the pyramid recurrent architecture contains the recurrent information with varied sizes. Then the low-dimension feature maps are directly resampled to get features of the same size via bilinear interpolation. Finally, different levels of recurrent features are concatenated to form the comprehensive feature representation, which carries both local and global context recurrent information. And a 1 × 1 convolutional layer is used to reduce the dimension of concatenated features. Finally, the representation is fed into a fully-connected layer to get the predicted coordinates at this pyramid. The predicted coordinates is calculated by $\mathscr{L}_1$ loss.

The *PRNet* is composed of the feature extraction part (*ResNet101*) and 3 *pyramid RNN architecture*. The number of pyramid level is set to 4. Each *pyramid RNN architecture* can predict coordinates, so the loss of *PRNet* is expressed as: $\mathscr{L}_{PR} = \mathscr{L}_{P1} + \mathscr{L}_{P2} + \mathscr{L}_{P3}$.

3.3 *FPRNet*

In order to make communications between *PRNet* and *FlowNet*, we propose a unified framework *FPRNet*, to jointly learn the coordinates of bounding box and optical flow. Therefore, the global loss is calculated by:

$$\begin{aligned} &\mathscr{L}(X) = \mathscr{L}_{PR}(X) + \lambda \mathscr{L}_f(X) \\ &\mathscr{L}_f(X_t, X_{t+1}) = \sum_{i,j}((u_{ij} - \delta_{u_{ij}})^2 + (v_{ij} - \delta_{v_{ij}})^2) \end{aligned} \tag{3}$$

where $\mathscr{L}_f$ is the endpoint error (EPE) loss [14]. And the λ is set to 0.5.

As is shown in Figure 2, the features are propagated between *PRNet* and *FlowNet* at different scales for the final predictions. In detail, features from *PRNet* are resized to match the features of *FlowNet* and both of the features are concatenated, vice versa. The two subnetworks are jointly learnt at feature space for characterizing useful feature representations (location and motion) at different scales.

4 Network Training and Implementation

In this section, we utilize the coordinates of the bounding box at a time by iteratively updating both *PRNet* and *FlowNet* and gradually optimize the target function.

4.1 Network Training

Offline Training The feature extraction part and *FlowNet* are initialized with [24] and [14], respectively. When training the *PRNet*, we fix the weights of the *FlowNet*, and train the network on the ILSVRC2015 [40]. We starts the learning rate from $1e-8$ and decreases it by half for every 10000 iterations and continue for approximately 650,000 iterations which takes roughly one week. On the contrary, *FlowNet* is trained with the Scene Flow dataset [34] by freezing the weights of *FlowNet*. To balance the weights between two subnetworks, we employ a λ in the combined loss.

Online Training for Tracking In the first frame, to adapt the model on a specific object for online tracking, we finetune the *PRNet* using the coordinates of bounding box in the first frame on each individual sequence.

4.2 Network Implementation

We utilize a modified version of the Caffe [28] framework. The inputs are the two consecutive frames (whole image) and the coordinates of bounding box in the previous frame. The input size is 854 × 480. In each pyramid RNN, the convolutional layers has 512 units. All of the fully-connected layers has 4 units. The ADAM gradient optimizer is employed with the default momentum and weight decay.

5 Experimental Results and Analysis

To evaluate the proposed *FPRNet* tracker, we tested the proposed method on 2 benchmark datasets: Object Tracking Benchmark (OTB) [45, 46] and TC128 [31] comparing with existing trackers.

We test our tracker on two OTB datasets: OTB50[45] which has 50 video sequences, and OTB100 [46] which has 100 video sequences including OTB50. And the OTB50 and OTB100 dataset include 50 and 100 sequences respectively tagged with 11 attributes. The tracking performance was measured by conducting a one-pass evaluation (OPE) based on two metrics: center location error and overlap ratio. The center location error measures the distance between the center of the tracked frame and the ground truth, and the bounding box overlaped ratio measures the Intersection-over-Union (IOU) ratio between the tracked bounding box and the ground truth.The TC128 dataset contains 128 color sequences with 11 challenge factor annotations.

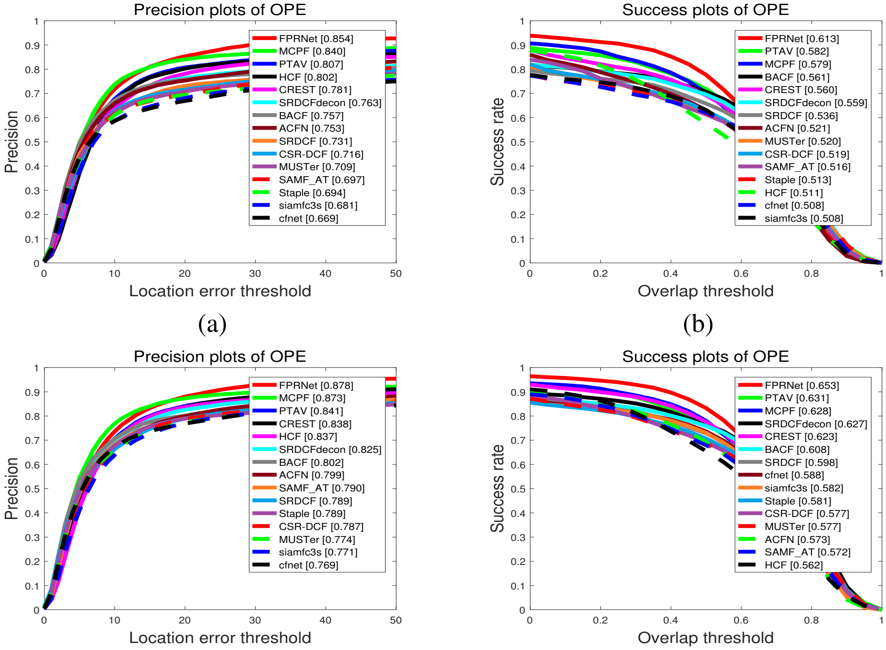
5.1 State-of-the-art Comparison

Experiments on OTB50 and OTB100 Dataset

We compare the proposed RRNet tracker on both OTB50 and OTB100 datasets with the following recent published 14 trackers: MCPF [50], PTAV [16], HCF [33], CREST [41], SRDCFdecon [12], BACF [17], ACFN [7], SRDCF [8], CSR-DCF [32], MUSTer [27], SAMF_AT [5], Staple [3], siamfc3s [4] and cfnet [44]. The OPE criteria on OTB is to evaluate our RRNet. The results are shown in Figure 5 (Left two images are the plots of precision and success rate on OTB50 dataset, and right two images are the plots of precision and success rate on OTB100 dataset). According to Figure 5, our *FPRNet* achieves the best performance among the state-of-the-arts on both datasets. The precision and the success plots are 0.854 and 0.613 on OTB50, and 0.878 and 0.653 on OTB 100, respectively.

Experiments on TC128 Dataset

The TC128 dataset is a challenging benchmark with 128 full color videos[31]. The 11 challenge factors for each sequence are also provided. A comparison with state-of-the-art trackers in precision plots is shown in Figure 6a. Among the compared methods, our



(a) (b)

(c) (d)

Figure 5: (a) and (b) are the precision and success plots on OTB50, respectively. (c) and (d) are the precision and success plots on OTB100, respectively.

approach improves the precision score from 0.7739 of the state-of-the-art tracker to 0.7777. The Figure 6b shows the success plot over all the 128 videos in the TC128 dataset. The *FPRNet* tracker outperforms state-of-the-art approaches with an AUC score of 0.5657.

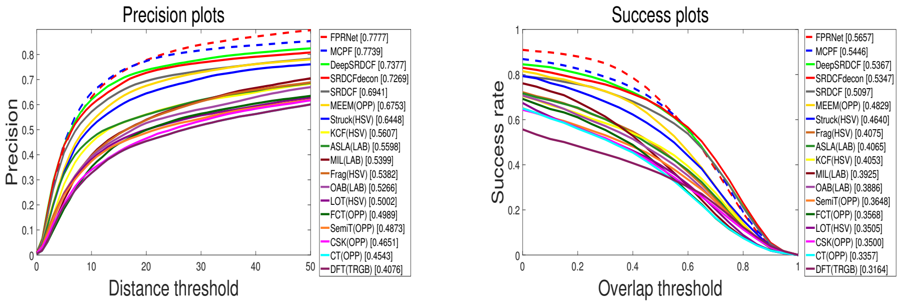
5.2 Attribute Analysis and Discussion

In the OTB100 dataset, different videos are annotated with 11 attributes, including fast motion (FM), background clutter (BC), motion blur (MB), deformation (DEF), illumination variation (IV), in-plane rotation (IPR), low resolution (LR), occlusion (OCC), out-of-plane rotation (OPR), out-of-view (OV), and scale variation (SV). Table 1 contains the different attributes success score on the OTB100 dataset. By considering the speed and attributes (the *FPRNet* tracker ranks top 1 on 10 out of 11 attributes in success plots) in Table 1, our *FPRNet* are more robust than the state-of-the-arts trackers.

According to the experimental results, the proposed *FPRNet* tracker achieves high performance. The possible reasons are listed as follows. (1) The *FPRNet* tracker exploits the spatial details and semantic information from different level of feature pyramid recurrent subnetwork, which can effectively process the frames containing the target with different appearance variances. (2) The comprehensive context-aware information effectively can handle the background clutters, in-plane rotations and out-of-plane rotations and other difficult cases, which can result in large appearance variance. (3) The jointly representations from two subnetworks can provide more useful spatiotemporal information for tracking.

Tracker	FPRNet	MCPF	PTAV	CREAST	ACFN	BACF
FM	0.613	0.608	0.582	0.606	0.517	0.518
BC	0.656	0.601	0.649	0.618	0.536	0.625
MB	0.622	0.573	0.590	0.608	0.556	0.563
DEF	0.656	0.620	0.640	0.664	0.611	0.596
IV	0.657	0.628	0.632	0.644	0.567	0.630
BC	0.656	0.601	0.649	0.618	0.536	0.625
IPR	0.621	0.598	0.580	0.599	0.515	0.556
LR	0.622	0.598	0.563	0.546	0.414	0.512
OCC	0.627	0.595	0.596	0.575	0.538	0.584
OPR	0.647	0.608	0.607	0.599	0.560	0.582
OV	0.585	0.553	0.570	0.566	0.494	0.525
SV	0.629	0.620	0.599	0.573	0.529	0.551
OverAll	0.653	0.628	0.631	0.623	0.573	0.608

Table 1: success scores of average AUC of the 6 state-of-the-art trackers under different attributes of test sequences in OPE on OTB100. (The red fonts indicate the best performance, the blue fonts indicate the second best ones and the green fonts marks the third best ones.)



(a) (b)

Figure 6: The precision plots and success plots on TC128 dataset are shown in (a) and (b).

6 Conclusions

An end-to-end tracking algorithm based on pyramid recurrent network (*PRNet*) and the *FlowNet*, which is referred to as *FPRNet*, is proposed in this paper. The *FPRNet* tracker predicts the coordinates of bounding box at different levels of pyramid with motion information from *FlowNet*. The *FPRNet* has achieved outstanding performance in 2 large public tracking benchmarks. Since our framework is a very flexible with great rooms for adding more pyramid RNNs and we will investigate to extend this work to design more efficient tracking algorithms in the future.

Acknowledgments: This work was supported in part by the Natural Science Foundation of China under Grant 61472102, and 61672194, by the Distinguished Youth Science Foundation of Heilongjiang Province of China under Grant JC2018021, by the Shandong Provincial Natural Science Foundation, China under Grant ZR2016FM04, and by the Humanity and Social Science Youth foundation of Ministry of Education of China under Grant 14YJC760001.

References

[1] Boris Babenko, Ming Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1619, 2011.

[2] Sean Bell, C. Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. pages 2874–2883, 2015.

[3] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip H. S. Torr. Staple: Complementary learners for real-time tracking. 38(2):1401–1409, 2015.

[4] Luca Bertinetto, Jack Valmadre, JoÃčo F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision*, pages 850–865, 2016.

[5] Adel Bibi, Matthias Mueller, and Bernard Ghanem. Target response adaptation for correlation filter tracking. pages 419–433, 2016.

[6] David S. Bolme, J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *Computer Vision and Pattern Recognition*, pages 2544–2550, 2010.

[7] Jongwon Choi, Hyung Jin Chang, Sangdoo Yun, Tobias Fischer, Yiannis Demiris, and Young Choi Jin. Attentional correlation filter network for adaptive visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4828–4837, 2017.

[8] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *IEEE International Conference on Computer Vision*, pages 4310–4318, 2015.

[9] Martin Danelljan, Gustav HÃd'ger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. 2015.

[10] Martin Danelljan, Gustav HÃd'ger, Fahad Shahbaz Khan, and Michael Felsberg. Convolutional features for correlation filter based visual tracking. In *IEEE International Conference on Computer Vision Workshop*, pages 621–629, 2015.

[11] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. pages 6931–6939, 2016.

[12] Martin Danelljan, Gustav HÃd'ger, Fahad Shahbaz Khan, and Michael Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. pages 1430–1438, 2016.

[13] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision*, pages 472–488, 2016.

[14] Alexey Dosovitskiy, Philipp Fischery, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. pages 2758–2766, 2015.

[15] Heng Fan and Haibin Ling. Sanet: Structure-aware network for visual tracking. pages 2217–2224, 2016.

[16] Heng Fan and Haibin Ling. Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In *IEEE International Conference on Computer Vision*, pages 5487–5495, 2017.

[17] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *IEEE International Conference on Computer Vision*, pages 1144–1152, 2017.

[18] Susanna Gladh, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Deep motion features for visual tracking. 2016.

[19] Helmut Grabner, Christian Leistner, and Horst Bischof. Semi-supervised on-line boosting for robust tracking. In *European Conference on Computer Vision*, pages 234–247, 2008.

[20] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *IEEE International Conference on Computer Vision*, pages 1781–1789, 2017.

[21] Bohyung Han, Dorin Comaniciu, Ying Zhu, and Larry S. Davis. Sequential kernel density approximation and its application to real-time visual tracking. *IEEE Trans Pattern Anal Mach Intell*, 30(7):1186–1197, 2008.

[22] Bohyung Han, Jack Sim, and Hartwig Adam. Branchout: Regularization for online ensemble tracking with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 521–530, 2017.

[23] Sam Hare, Amir Saffari, and Philip H. S. Torr. Struck: Structured output tracking with kernels. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 38(10): 2096–2109, 2016.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[25] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. pages 749–765, 2016.

[26] J. F. Henriques, R Caseiro, P Martins, and J Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[27] Zhibin Hong, Zhe Chen, Chaohui Wang, and Xue Mei. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. pages 749–758, 2015.

[28] Jia, Yangqing, Shelhamer, Evan, Donahue, Jeff, Karayev, Sergey, Long, and Jonathan. Caffe: Convolutional architecture for fast feature embedding. pages 675–678, 2014.

[29] Quoc V Le, Navdeep Jaitly, and Geoffrey E Hinton. A simple way to initialize recurrent networks of rectified linear units. *Computer Science*, 2015.

[30] Guanbin Li, Yuan Xie, Tianhao Wei, Keze Wang, and Liang Lin. Flow guided recurrent neural encoder for video salient object detection.

[31] P. Liang, E Blasch, and H. Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Image Processing*, 24(12):5630–5644, 2015.

[32] Alan Lukezic, Tomas Vojir, Luka Cehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. *International Journal of Computer Vision*, pages 1–18, 2016.

[33] Chao Ma, Jia Bin Huang, Xiaokang Yang, and Ming Hsuan Yang. Hierarchical convolutional features for visual tracking. In *IEEE International Conference on Computer Vision*, pages 3074–3082, 2015.

[34] Nikolaus Mayer, Eddy Ilg, Philip HÃd’usser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[35] Xue Mei and Haibin Ling. Robust visual tracking using âĎŞ1 minimization. In *IEEE International Conference on Computer Vision*, pages 1436–1443, 2010.

[36] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Computer Vision and Pattern Recognition*, pages 4293–4302, 2016.

[37] Hyeonseob Nam, Mooyeol Baek, and Bohyung Han. Modeling and propagating cnns in a tree structure for visual tracking. 2016.

[38] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, Jongwoo Lim, and Ming Hsuan Yang. Hedged deep tracking. In *Computer Vision and Pattern Recognition*, pages 4303–4311, 2016.

[39] David A. Ross, Jongwoo Lim, Ruei Sung Lin, and Ming Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3): 125–141, 2008.

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[41] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson W. H. Lau, and Ming Hsuan Yang. Crest: Convolutional residual learning for visual tracking. In *IEEE International Conference on Computer Vision*, pages 2574–2583, 2017.

[42] Shuyang Sun, Zhanghui Kuang, Wanli Ouyang, Lu Sheng, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. *arXiv preprint arXiv:1711.11152*, 2017.

[43] Ran Tao, Efstratios Gavves, and Arnold W. M. Smeulders. Siamese instance search for tracking. pages 1420–1429, 2016.

[44] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip H. S. Torr. End-to-end representation learning for correlation filter based tracking. pages 5000–5008, 2017.

[45] Yi Wu, Jongwoo Lim, and Ming Hsuan Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2411–2418, 2013.

[46] Yi Wu, Jongwoo Lim, and Ming Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 37(9):1834–1848, 2015.

[47] Xiaqing Xu, Bingpeng Ma, Hong Chang, and Xilin Chen. Siamese recurrent architecture for visual tracking. In *IEEE International Conference on Image Processing*, pages 1152–1156, 2017.

[48] Tianyu Yang and Antoni B. Chan. Recurrent filter learning for visual tracking. In *IEEE International Conference on Computer Vision Workshop*, pages 2010–2019, 2017.

[49] Peng Zhang, Tao Zhuo, Hanqiao Huang, and Mohan Kankanhalli. Saliency flow based video segmentation via motion guided contour refinement. *Signal Processing*, 142, 2017.

[50] Tianzhu Zhang, Changsheng Xu, and Ming Hsuan Yang. Multi-task correlation particle filter for robust object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4819–4827, 2017.

[51] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. pages 408–417, 2017.