Fine-Grained Motion Representation For Template-Free Visual Tracking

Kai Shuang[1,2] Yuheng Huang [1] Yue Sun[3] Zhun Cai[3] Hao Guo[3]
[1]State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing, China
[2]Science and Technology on Communication Networks Laboratory, Shijiazhuang, China
[3]Beijing Trusfort Technology Co., Ltd.
{shuangk,hyhlryf}@bupt.edu.cn,{sunyue,caizhun,guohao}@trusfort.com

Abstract

The object tracking task requires tracking the arbitrary target in consecutive video frames. Recently, several attempts have been made to develop the template-free models to attain generality. However, current template-free paradigm only estimates the displacement to approximate the motion of the object. The displacement is insufficient to represent complex bounding box transformation, including scaling and rotation. We argue that the coarse-grained representation of object motion limits the performance of current template-free approaches. In this paper, we explore the finer-grained motion estimation to improve the accuracy of the template-free model. In respect of the image space, our method estimates the transformation for each pixel in the image. Concern on the motion representation, we represent the motion by the transformation parameterized by displacement, scaling, and rotation. By applying the differential vector operators on the optical flow, our approach estimates both displacement, scaling, and rotation for each pixel in a unified theory. To the best of our knowledge, we are the first work to model the displacement, scaling and rotation in a unified theory with the optical flow. To further improve the localization accuracy, we develop the appearance branch to introduce the appearance information into our model. Furthermore, to suppress optical flow estimation failure samples during training, we propose a novel loss function Limited L1. The experiment shows our model ***FGTrack*** *achieves state-of-the-art performance on both NFS and VOT2017 datasets.*

1. Introduction

Generic visual object tracking task has been an essential task in computer vision, which is a hard problem. Given a template (or target) in an axis-aligned [44, 43, 30, 15, 31, 13] or rotated [24] bounding box in the first frame, the tracker is required to find the most similar patch in following frames. Visual object tracking is fundamental in a wide range of visual applications, such as automatic surveillance, self-driving, and augmented reality.

Figure 1. Visualization of the optical flow. In the visualization of the optical flow (right), different colours indicate different moving directions.

Requiring to track arbitrary objects, the generic visual object tracking is challenging in variations such as scale variation, deformation, and illumination variation. It is difficult or infeasible to learn such a robust feature from a single shot. To make up for this deficiency, template-based trackers have to develop online training methods [3, 8] to update models while tracking. To track objects without learning appearance features from the target, AMNet [21] proposed the template-free tracking paradigm. Template-free models believe that since the generic object tracking is required to track arbitrary objects, appearance features related to specific class should not dominate. Otherwise, it could limit the generalization ability of the model. Instead of learning a template from the target object, template-free approaches directly estimate the movement distance and direction of the object through the dense optical flow. The dense optical flow contains the direction and distance of movement of each pixel in the image. As shown in Fig.1, the optical flow can discriminate foreground from not only the non-semantic background but also semantic backgrounds with similar appearance, especially for moving objects. For stationary objects, the optical flow is zero. Thus

it is feasible to estimate the transform of either moving or stationary objects by the optical flow. Since the optical flow does not rely on any object-level information, the optical flow based tracker could be generalized to any object without online training.

However, most of the model-free approaches [21, 40, 49] only estimate the displacement of the object bounding box, which could not cope with the complex transformation of the bounding box, including scaling and rotation. Limited by poor movement representation, these methods provide poor accuracy with only displacement estimation. As a workaround, these models use the displacement information to guide a template-based discriminative model to find the target position. This paradigm leads the model to rely heavily on the template-based model, which violates the intention of the model design.

To overcome this restriction and drive the template-free models with better performance, we aim at predicting motion information in finer-grained with the optical flow by proposing *FGTrack*. Our fine-graininess is two-fold. As regards the image space, we estimate the bounding box transformation for each pixel in the image. Concerning motion representation, we represent the bounding box movement as the bounding box transformation parameterized by the displacement, scaling, and rotation. These three parameters are sufficient to track a moving rectangle bounding box with any two-dimensional transformation. Since the dense optical flow contains movement information of each pixel in frames, we believe that the dense optical flow is sufficient to estimate the complex transformation in pixel-level.

To achieve this goal, we introduce a cascaded tracking architecture based on a pixel-level transformer and an object-level reducer. The pixel-level transformer models the dense optical flow as a velocity field through the vector field theory and estimates the displacement, scaling, and rotation in a unified framework. We estimate these three transforms for each pixel neighborhood with the differential vector operators. To estimate the transform of the target bounding box, the object-level reducer aggregates the pixel-level motion information to bounding box transform parameters. Benefiting from the template-free design, our backbone architecture does not rely on class-specific features. By predicting the object movement in pixel-level, our model attains generality of the learned models and robustness.

While we gain the generality of the model from motion features, we also found that the appearance features are critical to improving the localization accuracy. In order to utilize the appearance features without affecting the generalization ability of the model, we design an auxiliary appearance feature branch to improve the performance for the optical flow branch. We introduce a spatial attention mechanism to help the object-level reducer focus on regions with an accurate estimation based on low-level appearance feature. To improving the localization accuracy, we extend the bounding box regression proposed by [16] through a rotation transform function and fusion features.

As described above, our model estimates the relative movement between two frames. Our framework is developed based on the optical flow estimation. Limited by current optical flow estimators, there are cases of estimated failures inevitably. Failure samples lead to a large loss, which causes the network difficult to converge. To solve the issue, we propose the Limited L1 Loss function. The Limited L1 loss suppresses the gradient of the outliers to help the model focus on the crucial gradients.

We summarized our main contributions in fourfold: (1) We propose a unified framework to estimate the bounding box displacement, scaling, and rotation with the optical flow. The framework utilizes the finer-grained motion information to improve the accuracy of the template-free tracking approaches. (2) We design a Limited L1 loss function to suppress the optical flow estimation failure samples as training the model. (3) We introduce the appearance information to the template-free approaches to improve the localization accuracy without losing the generalization ability of the model. (4) We demonstrate that our proposed FGTrack achieves state-of-the-art performance on NFS [15] and VOT2017 [23] benchmarks.

2. Related work

2.1. Motion features in video analysis

In this section, we briefly review approaches using motion features in video analysis. Motion feature is one of the essential features in video analysis, which usually considered as one of the approaches to introduce the temporal information into the model. The dense optical flow is widely used as the motion features in computer vision tasks.

A part of approaches considers the optical flow as a temporal feature representation [38, 36, 14]. [36] first propose a temporal-spatial CNN architecture with two branches, one for CNN features (spatial stream), the other for optical flow features (temporal stream). In the object tracking, SINT+ [38] employs the optical flow to filter out motion inconsistent candidates. [25] uses the optical flow by stacking the RGB image and optical flow as the input of the Siamese network.

Another part of approaches uses the optical flow as the movement estimation [40, 49, 47, 27, 39]. These approaches treat the optical flow as the absolute numerical value instead of the motion pattern of the object. [47, 49, 27] warp history features to the current features by optical flow for spatial consistency. [39, 40] provide a rough object movement estimation by the optical flow and guide the appearance model to locate the target position ac-

curately. Most optical flow based approaches only utilize the displacement information in the optical flow. In this paper, we further mine more information in the optical flow, including scaling and rotation.

2.2. Visual Object Tracking

In the last few years, a number of tracking approaches have been developed to handle various challenges in visual object tracking. With the great success of deep learning, CNN is a standard component of modern trackers. Follow-up works improve trackers in different aspects.

Several works focus on learning methods. [34] employs spatial attention by proposing a novel reciprocative learning approach. [41] introduces an unsupervised approach allowing the model to be trained on any video. DiMP [3] proposes several useful components to provide fast convergence at online training.

Other recent works focus on the representation of the tracking models. Considering the target-level representation, [28] integrates the target-aware features with Siamese networks. Towards segmentation-level representation, [42] proposes a unified approach to address both visual object tracking and visual object segmentation (VOS) in one model. In our approach, we are further pushing the representation of the tracking models to the pixel-level.

3. Proposed Method

The core of the visual object tracking task is to estimate the displacement, scaling, and rotation of the target bounding box. The visual tracking task will be solved correctly if the three parameters are well estimated. Previous work with template-free model only estimates the target translation by optical flow. To handle scale variations in complex sequences, most trackers [26, 10] use the pyramid method to search the best scale exhaustively. Although some efficient search methods [9] have been developed, the pyramid method is still time-consuming. Rotation, which is crucial for the high accuracy bounding box representation, is always ignored in most tracking approaches. We argue that the dense optical flow contains enough information to estimate all three transformation parameters since it contains the movement information for each pixel.

In the paper, we propose a novel tracking framework FGTrack predicting all three parameters with dense optical flow. FGTrack is composed of two branches: a movement estimation branch and an auxiliary appearance branch, as shown in Fig. 2.

The movement estimation branch contains an optical flow estimator as the backbone, a pixel-level transformer to estimate the movement for each pixel, and an object-level reducer to estimate the movement of the object. This branch is the main branch to estimate the object movement, which takes two consecutive frames as input. The optical flow estimator first extracts the optical flow information from the image pairs. Then the pixel-level transformer predicts the displacement, scaling, and rotation for each pixel in the target frame. The following Pooling layers are applied to extract the object region in the target frame. To predict the movement of the object, the object level reducer aggregates the changes for each pixel.

The movement estimation branch estimates the displacement, scaling and rotation for the target bounding box, which is sufficient to track the target position in the next frame. The estimation is entirely based on the dense optical flow information. However, even state-of-the-art optical flow estimators still fail in some hard cases, e.g., fast scene changing and objects with few features. We propose the appearance branch to improve the performance of our model. The appearance branch is designed as an auxiliary branch to improve feature representation and localization accuracy. To help the reducer focus on regions where pixel-level transformers perform well, we design a spatial attention module to weight the estimation for each spatial location. We also employ a refinement module to fine-tune the bounding box with bounding box regression [16]. With fusion features and the rotation regressor, the refinement module improves the accuracy of the bounding box localization.

The notations in this paper are described as follows. We use $z \in \mathbb{R}^{W \times H \times 3}$ and $X \in \mathbb{R}^{W \times H \times 3}$ to denote the first (or target) frame and the current frame, respectively. The superscript s denotes the search region of the specific frame, e.g., $X^S \in \mathbb{R}^{W^S \times H^S \times 3}$ means the cropped search region in the current frame. Each search region is cropped and padded to the fixed size. The superscript b indicates the area cropped according to the target bounding box.

3.1. Pixel-Level Transformer

In this section, we introduce three transformers to compute the displacement, scaling, and rotation for each pixel only with the optical flow. In order to make full use of the information in the optical flow, we model the optical flow as a vector field. By treating the optical flow as a vector field, we parameterize the displacement, scaling, and rotation with three differential vector operators over the vector field. Specifically, for a pair of images $\left(z^S, X^S\right)$, we first compute the optical flow $f \in \mathbb{R}^{W^s \times H^s \times 2}$. For the pixel at (u, v), the $f(u, v)$ is a 2-d vector (d_x, d_y), which represents the pixel displacement in x-direction and y-direction, respectively. It is natural to model the optical flow as a discrete vector field represented by a vector-valued function $\mathcal{F} : S \rightarrow \mathbb{R}^2$, where S is a subset of $\mathbb{Z}^2$. In the rest of the section, we describe three vector operators to compute displacement, scaling, and rotation.

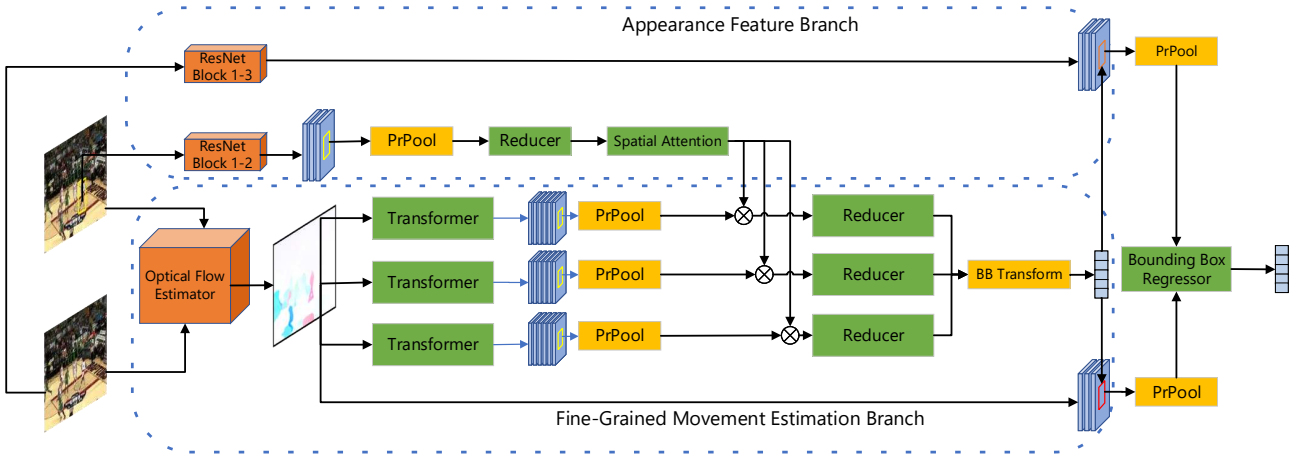

Figure 2. Detailed architecture of FGTrack. Blocks in orange are pre-trained by corresponding datasets, blocks in green are trained offline, blocks in yellow are parameter-free.

Figure 3. Visualization of zoom in and out of the object. The first column and the second column correspond to the first frame and the second frame, respectively. Best viewed in color.

Displacement The displacement is fundamental in tracking a moving object. According to the definition of the optical flow, the displacement at the position (u, v) can be inferred by optical flow itself:

$$\boldsymbol{d}_{t(u,v)} = \mathcal{F}(u, v) \tag{1}$$

where $\boldsymbol{d}_{t(u,v)} \in \mathbb{R}^2$ indicates the displacement of the pixel at (u, v).

Scaling We represent the scale changing as the relative changing of the width and the height of the bounding box, i.e.:

$$\boldsymbol{d}_s = (\mathrm{d}w, \mathrm{d}h) = \left(\frac{w_2}{w_1}, \frac{h_2}{h_1}\right) \tag{2}$$

In the pixel level, when the bounding box becomes larger, pixels of the object *outflow* through the bounding box (Fig. 3); when the bounding box becomes smaller, pixels of the object *inflow* through the bounding box. The field theory describes this phenomenon as flux. The flux describes the effect (e.g., magnetic field, fluid) passes through a surface (in three-dimension case) or a curve (in two-dimension case). We introduce the flux of the bounding box $\boldsymbol{I}_{bb}$ to represent the pixels pass through the bounding box. Intuitively speaking, the more pixels pass through the bounding box, the smaller the bounding box becomes:

$$\boldsymbol{I}_{bb} \propto \boldsymbol{d}_s \tag{3}$$

To estimate the scale changing in pixel level, we estimate the scale transform for each pixel by the differential of the flux, i.e. , the *divergence*. Thus, the scale changing at the region near the position (u, v) is related to the divergence of the optical flow field [33]:

$$\boldsymbol{d}_{s(u,v)} = \phi_s(\operatorname{div} \mathcal{F})(u, v) \tag{4}$$

For the two-dimension continuous vector field $F(x, y) = \langle F_x, F_y \rangle$, the divergence, if it exists, is given by:

$$\operatorname{div} F = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} \tag{5}$$

A simple way to compute the numerical gradient in the discrete domain is to calculate the central difference for interior data points [4]. For the single variable function $g(x)$, the discrete gradient at $x = x_i$ is approximated by:

$$\left.\frac{\mathrm{d}g}{\mathrm{d}x}\right|_{x=x_i} = \frac{g\left(x_{i+1}\right) - g\left(x_{i-1}\right)}{2} \tag{6}$$

We employ a convolutional neural network to parametrize ϕ_s. Detailed implementation is described in Sec. 4.

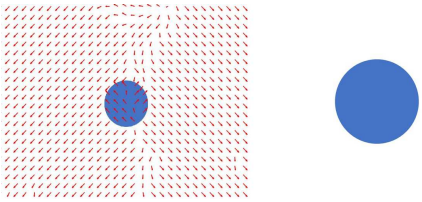

Figure 4. A failure case of the optical flow estimation. The left frame is the first frame, and the right frame is the second frame. The size and direction of the arrow in the left indicate the displacement of each pixel. The optical flow is estimated by the PWC-Net [37].

Rotation We represent a rotated rectangle as the angle between the edge of the rectangle and the horizontal line. The rotation is formulated as:

$$d_\theta = \theta_{t+1} - \theta_t \tag{7}$$

To find the rotation pattern of the pixel, we introduce the *circulation* around the bounding box boundaries. The circulation is the amount of movement directions that pushes along a closed boundary or path, which represents how pixels may rotate. We estimate the rotation of the bounding box d_θ by the circulation around the bounding box Γ_{bb}:

$$\mathbf{\Gamma}_{bb} \propto d_\theta \tag{8}$$

We estimate the rotation for each pixel with the differential of the circulation, i.e., the curl. Our model employs the curl to estimate the rotation of the region around the pixel at (u, v):

$$d_{\theta(u,v)} = \phi_\theta(\operatorname{curl} \mathcal{F})(u, v) \tag{9}$$

For the two-dimensional field $F(x, y) = \langle F_x, F_y \rangle$, the curl is given by:

$$\operatorname{curl} F = \frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \tag{10}$$

where the gradient is approximated as Eq. 6. We also employ a convolutional neural network to parametrize the ϕ_θ, as described in Sec. 4.

3.2. Appearance Feature Branch

Object Level Reducer with Spatial Attention Pixel level transformers estimate the displacement, scaling, and rotation for each pixel. The reducer aggregates the pixel-level information to object-level to locate the object. We first extract the patch of the first frame by Precise ROI Pooling (`PrROI`) [20]. Then, we design object-level reducers γ_{x} to aggregate the movement in pixel $\boldsymbol{D}_x \in \mathbb{R}^{w^b \times h^b \times 2}$ to object-level movement $\boldsymbol{d}_{x_{\text{obj}}}$:

$$\boldsymbol{d}_{\text{x}_{\text{obj}}} = \gamma_{\text{x}}\left(\boldsymbol{D}_{\text{x}}\right) \tag{11}$$

where w and h is the width and height of the region of the interest, and x can be t, s, and θ which denotes displacement, scaling and rotation, respectively.

Limited by the performance in current optical flow estimator, the optical flow estimation is not accurate, especially in regions lacking features. Fig. 4 shows an example of estimation failure. In the example, the estimator fails to estimate the movement of the texture-free circle with the pure background correctly, due to the aperture problem [32]. We alleviate this issue by introducing a spatial attention mechanism to enhance the weight around the feature-rich region and suppress the weight around the feature-free neighborhood. We first extract the appearance feature ω by a pre-trained feature extractor. Since the optical flow is the low-level feature, we extract features from the shallow layer (e.g., `layer2` in ResNet [18]) to consistent with the optical flow feature. Then we adopt a mini attention network ϕ_{att} to generate the attention map, followed by a normalization function ϕ_{norm}. Then spatial attention map is given by:

$$\omega^{\text{S}} = \phi_{\text{norm}}\left(\phi_{\text{att}}(\omega)\right) \tag{12}$$

Discriminative networks usually use the Softmax as the normalization function. This normalization ensures that the sum of the components of the output is 1. The feature map in the discriminative networks represents the pattern found in the input, which is quite different from the matrix in our method. Each value in the $\boldsymbol{d}_{\text{x}}$ has a absolute numerical meaning, since the $\boldsymbol{d}_{\text{x}}$ in our method represents the displacement, scaling or rotation for each pixel. In order to preserve the numerical information in the matrix, we design a normalization function:

$$\phi_{\text{norm}} = \tanh(\alpha \cdot \operatorname{Softmax}(x) - \beta) \tag{13}$$

The tanh function reduces the small value to 0 while keeping the larger value in the $\boldsymbol{d}_x$. By introducing the spatial attention mechanism, the object-level movement is computed as:

$$\boldsymbol{d}_{\text{x}_{\text{obj}}} = \gamma_{\text{x}}\left(\boldsymbol{D}_{\text{x}} \odot \omega^S\right) \tag{14}$$

Feature Fusion To improve the localization accuracy, we adopt a bounding box regression module with fusion features. By exploiting both the motion features and appearance features, our model gains a better representation for the regression module. We obtain the feature map by first concatenating them in the channel dimension, then applying a 1×1 convolutional layer to generate the fusion feature map. We always crop an axis-aligned proposal even if $\theta \neq 0$.

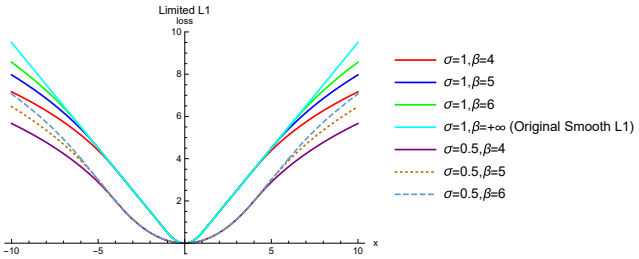

Figure 5. The Limited L1 loss with different hyper-parameters.

3.3. Limited L1 Loss

As described above, our model predicts the relative transformation of objects between two frames. In most videos, the movement of objects between adjacent frames is usually slight. There is an extreme imbalance between massive displacement and small displacement during training. Since the image pairs with relatively large displacement are rarely in the training data, the network trends to predict the small movement for all samples. Moreover, image pairs with relatively large displacement are particularly valuable as they contain rich motion information. A natural solution to the imbalance issue is to tune the gap between frames at sampling. However, it is tricky to choose the best sample gap. For a too large frame gap, the optical flow estimator fails as displacements get too large, which causes the model to be difficult to converge. Consider suppressing outliers when using a large frame gap, we propose the Limited-L1 loss. We introduce the Limited-L1 starting from briefly reviewing the smooth-L1 function.

The smooth-L1 loss [16] combines the L1-loss and L2-loss. Specifically, it behaves as L1-loss when the estimation error is high, and it behaves like L2-loss when the estimation error is close to zero. The hyper-parameter α controls the boundary between the L1 loss and L2 loss, which is usually taken as 1. We plot the initial smooth L1 loss in Fig. 5.

In our approach, the optical flow estimation failure leads to a large loss. Since our model does not update parameters in the optical flow estimator, failure samples are harmful and unhelpful in learning the representation of the vector field of optical flow. To suppress the noise from these outliers, we propose the Limited L1 loss function. The core idea of the Limited L1 loss is to limit the regression gradients from outliers (optical flow estimation failed samples). Based on this idea, we design a suppressed gradient formulation as:

$$\frac{\mathrm{d}L_{\text{limit}}}{\mathrm{d}x} = \begin{cases} \sigma^2 x, & |x| < \frac{1}{\sigma^2} \\ 1, & \frac{1}{\sigma^2} \le |x| < \beta \\ \frac{1}{x}, & |x| \ge \beta \end{cases} \tag{15}$$

By integrating the gradient in Eq.15 and meet the first-order continuous condition, we derive the Limited L1 loss:

block	output size	*backbone*
conv1	112×112	7×7, 64, stride 2
conv2_x	56×56	3×3 average pooling, stride 2 $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 128 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 128 \end{bmatrix}\times3$

Table 1. The network architecture of transformers.

$$L_{\text{limit}}(x,\sigma,\beta) = \begin{cases} 0.5\sigma^2 x^2, & |x| < \frac{1}{\sigma^2} \\ |x| - \frac{1}{2\sigma^2}, & \frac{1}{\sigma^2} \le |x| < \beta \\ \beta \ln |x| - \beta \ln \beta + b - 1, & |x| \ge \beta \end{cases} \tag{16}$$

Fig. 5 shows the plot of the Limited L1 loss with different β and σ. We use the hyper-parameter σ to control the boundary between the L1 function and L2 loss function. When the relatively larger σ is applied, the gradient of larger movement is promoted to learn the motion representation. The hyper-parameter β is tuned to suppress the loss of optical flow estimation failure samples.

3.4. Training

Our training data comes from TrackingNet [31] and ImageNet-VID [11], including the training set and validation set. To train the rotation head, we train our model on YouTube-VOS [45]. We use the minimum bounding rectangle (MBR) of the binary masks in the YouTube-VOS as the rotated bounding box. For each frame, the search region is cropped around the ground truth with 5 padding. The images pairs are sampled from the videos with a maximum gap of 15 frames. The ResNet-18 in our model is pre-trained by ImageNet [11], and MS-COCO [29]. The FlowNet2 is pre-trained according to the method in [19]. The cropped patches are resized to 224×224 before feature extraction. As describe in Sec. 3.3, we use Limited L1 Loss as the loss function, with $\beta = 5$ and $\sigma = 1.2$. We apply ADAM with the initial learning rate of 10^{-4} and using a factor of 0.25 decay every 10 epochs. We perform 50 epochs in training with 16 image pairs per batch.

4. Experiment

4.1. Implementation Details

For the optical flow estimator, we follow the implementation of FlowNet2 [19]. The discussion of different optical flow estimator is in Sec. 4.3. The calculation of divergence and the curl follows the Eq. 5 and Eq. 10. In our model, we

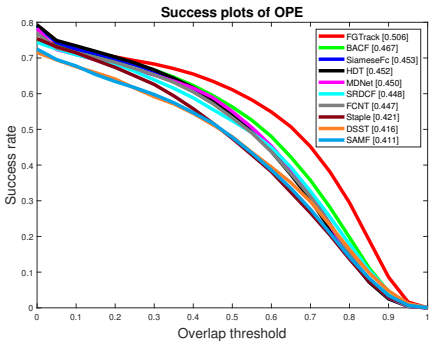

Figure 6. Success plots on the NFS [15] benchmark for comparison.

	FGTrack	C-COT[10]	ECO[7]	SiamFC[2]	DaSiamRPN[48]
AUC	**50.6**	49.2	47.0	45.3	39.5

Table 2. Comparison results with state-of-the-art trackers on the NFS benchmark.

apply the mean subtraction on the optical flow before computing the divergence and curl, which is omitted in Eq. 1 and Eq. 9 for simplicity. The mean subtraction reduces the effect of global motion, which helps the model learn the displacement-invariant scaling and rotation patterns. We design transformers following the ResNet [18], as shown in Table 1. We adopt one convolutional layer ($1\times1\times128$) for reducers, followed by two fully connected layers. The hidden unit of FC layers is 1000, and the final output dimension is 2, 2, 1 for displacement, scaling, rotation, respectively. Different transformers and reducers do not share parameters. For datasets with axis-aligned annotations (e.g., NFS [15], OTB [44, 43]), we design a rotation-free variant FGTrack (FGTrack-*rf*). The FGTrack-*rf* does not calculate the curl for the optical flow. In the appearance feature branch, we adopt a modified ResNet-18 as backbone extractor following the implementation in [42]. We extract the appearance features by the output of ResNet `block2` and feed them to an small attention network. This network composed of three convolutional layers ($1\times1\times128$, $3\times3\times128$, $1\times1\times128$). Then the attention map takes channel-wise summation before the element-wise multiplication with the feature map.

We implemented the FGTrack using PyTorch on NVIDIA GeForce 1080Ti GPU. The average speed of the tracker is 19.6 FPS, which is evaluated on the VOT2017 benchmark.

4.2. Results

Need For Speed [15] The Need For Speed (NFS) dataset consists of 100 videos from real-world scenarios. The annotations of the NFS are axis-aligned bounding boxes for all frames. The benchmark evaluates trackers with the average success rate at different thresholds.

We compare our tracker with state-of-the-art trackers. Fig. 6 and Table 2 reports the success plot and the AUC scores over all 100 videos. It is worth noting that the dataset is captured with a higher frame rate (240 FPS) cameras. The higher frame rate allows our approach to estimate the optical flow more accurately since the optical flow estimator gives a lousy performance as displacements get too large. Fig. 6 shows that our approach outperforms BACF with a relative gain of 4%. In the Table 2, we compare our trackers with state-of-the-art Siamese trackers and Correlation Filter trackers, including DaSiamRPN [48], C-COT [10], and ECO [7]. As discussed in [15], complex methods based on deep networks provide poor results on high frame videos. As a Siamese network based approach, DaSiamRPN only achieves 39.5 on AUC score. Meanwhile, C-COT and ECO, both based on correlation lters, achieve 47.0 and 49.2, respectively. Our approach outperforms all of them by achieving 50.6 of AUC score.

VOT Benchmark [24] VOT dataset, containing 60 videos, is a challenging dataset. The benchmark evaluates trackers with the Expected Average Overlap (EAO) metric. The dataset also adopts accuracy and robustness for evaluations. We conduct experiments on both VOT2016 [22] and VOT2017 [23] benchmark.

We evaluate the proposed FGTrack with state-of-the-art trackers. Table 3 reports the evaluation results on both VOT2016 and VOT2017 benchmark. The result indicates that our tracker shows competitive performance on the VOT2017 benchmark. Our approach is slightly better than another template-free method AMNet [21] (0.7% on EAO). The DaSiamRPN, which additionally uses large scale dataset YouTube-BB [35], ImageNet DET [11], performs better than our approach. On VOT2016 benchmark, our tracker shows the competitive robustness. Our approach outperforms FlowTrack [49] by 1.6% on EAO. As we discussed in Sec. 3, the optical flow estimator might fail in some cases, which leads to low accuracy of our tracker. We believe the accuracy of our model could be improved by a better optical flow estimator, as discussion in Sec. 4.3.

4.3. Ablation Study

In this section, we illustrate ablation studies to demonstrate the impact of each component in the proposed method. We conduct experiments on VOT 2017 benchmark. We adopt EAO [23] metric to evaluate our tracker.

Impact of backbone network We compare different backbone choice of FGTrack, including the appearance feature extractor and the optical flow estimator. The results are shown in Table 4. For optical flow estimator, we evaluate the performance on three different optical flow estimators: FlowNet [12], FlowNet2 [19] and PWC-Net [37]. On MPI

		DaSiamRPN[48]	**FGTrack**	FlowTrack[49]	CFCF[17]	AMNet[21]	ECO[7]	CCOT[10]	MCPF[46]	CRT[6]	ECO-HC[7]	Staple[1]
VOT2017	EAO ↑	0.326	0.287	-	0.286	0.28	0.28	0.267	0.248	0.244	0.238	0.169
	Accuracy ↑	0.56	0.47	-	0.509	0.48	0.483	0.494	0.51	0.463	0.494	0.53
	Robustness ↓	0.34	0.294	-	0.281	0.218	0.276	0.318	0.427	0.337	0.435	0.688
VOT2016	EAO ↑	0.411	0.35	0.334	0.390	-	0.375	0.331	-	-	0.322	0.295
	Accuracy ↑	0.61	0.55	0.578	0.54	-	0.55	0.538	-	-	0.54	0.54
	Robustness ↓	0.22	0.21	0.241	-	-	0.20	0.24	-	-	0.30	0.38

Table 3. Comparison with state-of-the-art trackers on the VOT2017 and VOT2016 benchmark.

	FlowNet ResNet-18	FlowNet 2 ResNet-18	PWC-Net ResNet-18	FlowNet2 ResNet-50
EAO ↑	0.261	0.287	0.294	0.285
Robustness ↓	0.33	0.294	0.278	0.305
Accuracy ↑	0.45	0.47	0.47	0.46

Table 4. Comparison of the different backbone on the VOT 2017 benchmark.

	Init	+Loss	+R	+BBR	+SA
EAO	0.248	0.253	0.265	0.279	0.287

Table 5. Analysis of the impact of the motion-feature-only model with smooth-L1 (Init), Limited L1 Loss (+Loss), rotated bounding box estimation (+R), bounding box regression (+BBR) and Spatial Attention (+SA).

Sintel Final Benchmark [5], these three estimators achieve EPE (the lower the better) of 8.81, 6.016, 5.042, respectively. The model with the FlowNet as the optical flow estimator performs an EAO of 0.261. The FlowNet2 achieve a significant gain of 0.026 in EAO score. The PWC-Net gives a EAO improvement to 0.294. The experiment indicates that the model can benefit from a more accurate optical flow estimator. Our model should be able to achieve higher performance by using a more accurate optical flow estimator. To compare different appearance feature backbone in our model, we perform the experiments on ResNet-18, ResNet-50. We observe that the deeper network does not bring significant improvements to the model. This phenomenon is not surprising since the appearance model acts as an auxiliary branch in our design.

Analysis of components To investigate the impact of the appearance feature branch and the Limited L1 Loss function, we train four variants of our model. **Init**: The baseline only uses the motion branch to predict the target position, which removes the whole appearance feature branch. Thus, this model only uses the motion feature to estimate the final result without the spatial attention and the bounding box regression. We use the smooth-L1 loss as training the baseline model. The Init model only predict the axis-aligned bounding box. **+Loss**: To investigate the impact of the proposed loss function, we replace the smooth-L1 loss to Limited L1 Loss. **+R** To disentangle the effect of the rotation bounding box, this model predict the rotated bounding box instead of the axis-aligned bounding box. **+BBR**: In this approach, we add the bounding box regression to the +R model. The bounding box regression uses the fusion features to estimate the transform of the displacement, scaling, and rotation. **+SA**: We add the spatial attention mechanism based on +BBR approach. The spatial attention mechanism helps transforms focus on the feature-rich region. We train all networks in the same settings.

Table 5 shows the results of these approaches. Without the appearance feature and the Limited L1 Loss, the baseline model Init achieves the EAO score of 0.248. The Limited L1 Loss provides a gain of 0.5% of EAO. The rotated bounding box representation gives 1.2% EAO improvement, which illustrates rotation estimation with curl field can advance the performance of the tracker. By applying the bounding box regression with fusion features, the model obtains a substantial increase of 1.4% in EAO score.

5. Conclusions

In this paper, we propose a fine-grained tracking framework based on template-free approach. Our tracking framework estimates the pixel-level movement for each frame without pixel-level annotation. By applying differential operators on the optical flow, we predict displacement, scaling, and rotation for each pixel in the target frame. We simultaneously design an auxiliary appearance feature branch to improve the localization accuracy without losing the generalization ability of the model. We further propose the Limited L1 Loss to suppress outliers during training. We performed comprehensive experiments on several tracking benchmarks. Our approach FGTrack achieves state-of-the-art performance in NFS and VOT2017 benchmark. By providing a pixel-level motion representation, our fine-grained approach demonstrates the potential of optical flow in visual tracking and could be applicable in many other tasks with the optical flow. We believe the accuracy of our approach could be further improved by a more powerful optical flow estimator and the integration with other tracking approaches, e.g. Siamese trackers.

Acknowledgement This work is supported by National Key Research and Development Program of China (2016QY01W0200), the open project of Science and Technology on Communication Networks Laboratory (614210403070617) and National Natural Science Foundation of China (U1534201).

References

[1] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr. Staple: Complementary learners for real-time tracking. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1401–1409, June 2016.

[2] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. Fully-convolutional siamese networks for object tracking. In G. Hua and H. Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 850–865, Cham, 2016. Springer International Publishing.

[3] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte. Learning discriminative model prediction for tracking. *arXiv preprint arXiv:1904.07220*, 2019.

[4] R. L. Burden and J. D. Faires. *Numerical Analysis*. The Prindle, Weber and Schmidt Series in Mathematics. PWS-Kent Publishing Company, Boston, fourth edition, 1989.

[5] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012.

[6] K. Chen and W. Tao. Convolutional regression for visual tracking. *IEEE Transactions on Image Processing*, 27(7):3611–3620, July 2018.

[7] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6931–6939, July 2017.

[8] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Atom: Accurate tracking by overlap maximization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[9] M. Danelljan, G. Hauml;ger, F. Shahbaz Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[10] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 472–488, Cham, 2016. Springer International Publishing.

[11] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[12] A. Dosovitskiy, P. Fischer, E. Ilg, P. Husser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, Dec 2015.

[13] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941, June 2016.

[15] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1134–1143, Oct 2017.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, June 2014.

[17] E. Gundogdu and A. A. Alatan. Good features to correlate for visual tracking. *IEEE Transactions on Image Processing*, 27(5):2526–2540, May 2018.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[19] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, July 2017.

[20] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[21] X. Jiang, P. Li, X. Zhen, and X. Cao. Model-free tracking with deep appearance and motion features integration. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 101–110, Jan 2019.

[22] M. Kristan et al. The visual object tracking vot2016 challenge results. In G. Hua and H. Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 777–823, Cham, 2016. Springer International Publishing.

[23] M. Kristan, A. Leonardis, et al. The visual object tracking vot2017 challenge results. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1949–1972, 2017.

[24] M. Kristan, J. Matas, A. Leonardis, T. Voj, R. Pflugfelder, G. Fernndez, G. Nebehay, F. Porikli, and L. ehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2137–2155, Nov 2016.

[25] L. Leal-Taix, C. Canton-Ferrer, and K. Schindler. Learning by tracking: Siamese cnn for robust target association. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 418–425, June 2016.

[26] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, June 2018.

[27] X. Li and C. C. Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, edi-

tors, *Computer Vision – ECCV 2018*, pages 93–110, Cham, 2018. Springer International Publishing.

[28] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang. Target-aware deep tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[30] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for uav tracking. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2016.

[31] M. Müller, A. Bibi, S. Giancola, S. Al-Subaihi, and B. Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *European Conference on Computer Vision*, 2018.

[32] K. Nakayama and G. H. Silverman. The aperture problemii. spatial integration of velocity information along contours. *Vision Research*, 28(6):747–753, 1988.

[33] R. C. Nelson and J. Aloimonos. Obstacle avoidance using flow field divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1102–1106, Oct 1989.

[34] S. Pu, Y. Song, C. Ma, H. Zhang, and M.-H. Yang. Deep attentive tracking via reciprocative learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 1935–1945, USA, 2018. Curran Associates Inc.

[35] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7464–7473, July 2017.

[36] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 568–576. Curran Associates, Inc., 2014.

[37] D. Sun, X. Yang, M. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, June 2018.

[38] R. Tao, E. Gavves, and A. W. M. Smeulders. Siamese instance search for tracking. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1420–1429, June 2016.

[39] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7942–7951, 2019.

[40] J. Wang, Y. He, X. Wang, X. Yu, and X. Chen. Prediction-tracking-segmentation. *CoRR*, abs/1904.03280, 2019.

[41] N. Wang, Y. Song, C. Ma, W. Zhou, W. Liu, and H. Li. Unsupervised deep tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[42] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr. Fast online object tracking and segmentation: A unifying approach. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[43] Y. Wu, J. Lim, and M. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, Sep. 2015.

[44] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[45] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[46] T. Zhang, C. Xu, and M. Yang. Multi-task correlation particle filter for robust object tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4819–4827, July 2017.

[47] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei. Flow-guided feature aggregation for video object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 408–417, Oct 2017.

[48] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu. Distractor-aware siamese networks for visual object tracking. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[49] Z. Zhu, W. Wu, W. Zou, and J. Yan. End-to-end flow correlation tracking with spatial-temporal attention. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 548–557, June 2018.