Robust Instance Tracking via Uncertainty Flow

Jianing Qian

CMU-RI-TR-20-31

July 9, 2020

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
David Held
Deva Ramanan
Leo Keselman

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

Abstract

Current state-of-the-art trackers often fail due to distractors and large object appearance changes. In this work, we explore the use of dense optical flow to improve tracking robustness. Our main insight is that, because flow estimation can also have errors, we need to incorporate an estimate of flow uncertainty for robust tracking. We present a novel tracking framework which combines appearance and flow uncertainty information to track objects in challenging scenarios. We experimentally verify that our framework improves tracking robustness, leading to new state-of-the-art results. Further, our experimental ablations shows the importance of flow uncertainty for robust tracking.

Acknowledgments

First, I would like to express my deepest appreciation to my advisor Professor David Held for his support over the past two years. His great insight in computer vision guide me through many obstacles I encountered during this project. More importantly, his passion and perseverance in robotics research enlighten and motivate me to become a better researcher.

I would also like to extend my deepest gratitude to my collaborators, Junyu Nan, Siddharth Ancha and Brian Okorn. They provide me with invaluable insights and support during this project. I would also like to thank Professor Deva Ramanan and Leo Keselman for being my committee member. Besides, I want to thank all the amazing people I met at the Robotics Institute.

Lastly, I want to thank my parents for their understanding and support of every decision I've made so far. Without them, I wouldn't be able to be here today.

Funding

This work was supported by the CMU Argo AI Center for Autonomous Vehicle Research.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

Instance tracking is an important task in video applications, such as autonomous driving, sports analytics, video editing, and video surveillance. In single-object tracking, the position of the target instance is given in the first frame of a video sequence; tracking algorithms need to predict the position of the same instance in each of the following frames.

Most state-of-the-art tracking methods use a convolutional neural network to extract features from the target object and features from the scene [27, 35, 38]. These methods use an approach of "tracking-by-one-shot-detection" [27]: a network is trained to match the appearance between an image of the target object and an image of the same object in the current frame.

Although this "tracking-by-one-shot-detection" approach has achieved impressive performance, it is prone to errors due to distractors and object appearance changes. First, if there are similar-looking objects in the video ("distractors"), tracking-by-one-shot-detection methods often switch to a distractor object (see Figure 1.1b); common examples include different objects of the same category or objects of similar color or texture. Likewise, if the object changes its appearance due to object deformations, image blur (from large camera or object motion), lighting changes, or other variations, tracking-by-one-shot-detection methods often lose track of the target object (see Figures 1.1a, 1.1c).

When there are distractor objects or large appearance changes, matching the object appearance alone will likely be insufficient for robust tracking. Instead, the

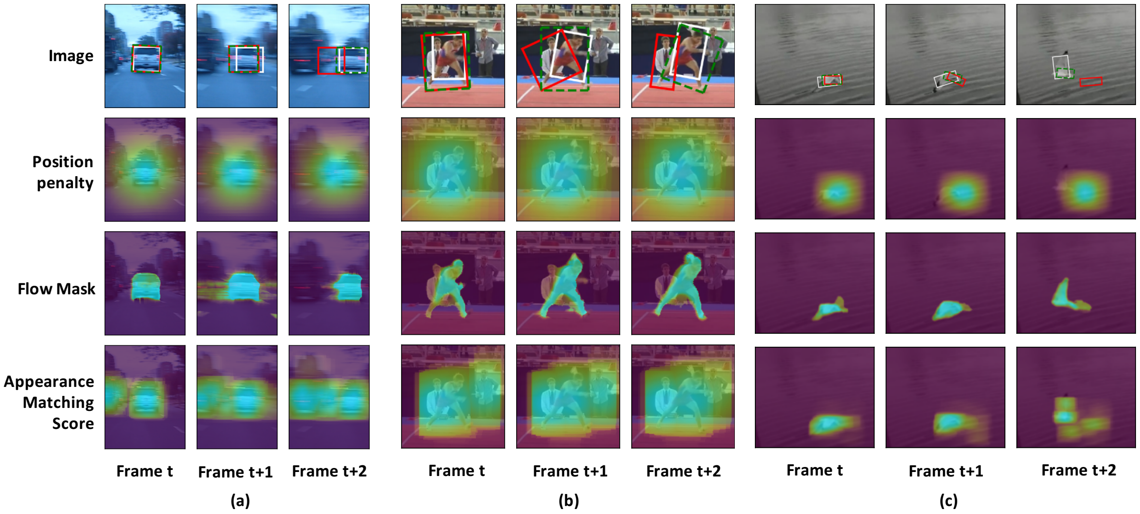

Figure 1.1: Three example errors that our method fixes (a): Failure case of baseline due to large camera motion; (b): Failure case of baseline due to distractors; (c): Failure case of baseline due to large motion of instance being tracked. In this figure, white boxes represent ground-truth boxes, red boxes represent predictions by SiamMask [35], green boxes show the results from our method.

tracker should make use of the tracked object's position. By tracking the position of the object throughout the video, the tracker can determine which object is the target and which are the distractors.

Past trackers have typically incorporated relatively weak position information to try to resolve these issues. For example, many methods [2, 27, 35, 38] use a position penalty that gives a lower score to bounding boxes that are farther away from the location of the detected object in the previous frame. However, a position penalty that is too strong will lose track of fast moving objects or objects under large camera motion; a position penalty that is too weak will not achieve the desired effect of ignoring distractors or handling large object appearance changes.

To address these issues, we explore how trackers can incorporate object position information in a more robust manner. Specifically, we use dense optical flow correspondences to track the position of the target object from one frame to the next. Dense optical flow methods jointly track the motion of the target object as well as the distractors or other nearby objects, and can thus be used to more robustly ignore distractor objects. Optical flow can also track objects over large appearance changes by reasoning about the position of the target object relative to the rest of the scene.

However, methods for dense optical flow can also make mistakes; incorporating such erroneous information can cause the tracking performance to degrade. Our main insight is that, to avoid such situations, we should use an estimate of optical flow uncertainty [17] to reason about our confidence in the optical flow estimate. We develop a novel probabilistic framework that estimates a tracking score for each bounding box based on the optical flow estimates and their uncertainties. These flow scores are combined with object appearance scores to estimate the new position of the tracked object.

We demonstrate that our method significantly improves tracking performance, when evaluated on the VOT 2016, 2018, and 2019 datasets, compared to the performance of the base tracker that our method builds upon. Our method is general in that our flow uncertainty tracking scores can be incorporated into any base tracker. We show that our method improves tracking robustness under distractors and large object appearance changes. Our contributions include

- A novel end-to-end differentiable method for combining segmentation and flow uncertainty for tracking
- Experimental demonstrations that our method outperforms current state of the art trackers
- Ablations showing the importance of each component of our method, especially flow uncertainty, for optimal tracking performance

Chapter 2

Related Work

2.1 2D Instance Tracking

Since the work of Bolme [3], correlation filter has been a popular approach for instance tracking. The method trains a filter online and tracks the target by correlating the filter over a search window. Significant efforts has been devoted to improve the performance, such as by learning a multi-channel filter [9, 13], integrating multi-resolution deep feature maps [6, 31] and mitigating boundary effects [5, 10]. Recently, instead of learning discriminative filter online, offline learning methods, especially siamese networks [2, 8, 12, 27, 35, 38], have considerably improved performance on 2D instance tracking by using a one-shot detection framework.

In order to track objects temporally, most trackers incorporate a position penalty to prevent large changes in position from one frame to the next. This can be achieved using a cosine window penalty [2, 27, 35, 38] or a Gaussian penalty [28]. Another approach to incorporate position information more implicitly is to input a search region cropped around the location of the tracked object in previous frame [2, 12, 27, 35, 38] or to restrict feature correlation to a local neighborhood [8]. Many previous works take the Bayesian approach for instance tracking, using a Kalman filter [1, 19, 36] or particle filter [15, 18, 29] to smooth the tracker output over time. To make the method more robust to distractors, DaSiamRPN [38] proposes a distractor-aware module to perform incremental learning during inference time. We show that our approach to avoiding distractors significantly outperforms these approaches.

Our tracker makes use of a segmentation mask of the tracked object from the previous frame. To obtain this mask, we use SiamMask [35], which achieves the state-of-the-art tracking performance. We combine this mask with uncertainty-aware optical flow to improve tracking performance in the face of distractors and large appearance changes.

2.2 Optical Flow

Optical flow has been widely used for video analysis and processing. Traditional methods for optical flow estimation includes variational approaches [14], possibly combined with combinatorical matching [32]. Recently, deep learning based methods [7, 16] have obtained state-of-the-art performance for optical flow estimation. Optical flow has been used to guide feature warping to improve performance of class-level object detection in videos [39]. Other work [34] uses optical flow to identify temporal connections throughout videos, and jointly updates object segmentation with flow models. In contrast to these applications, we use optical flow to improve tracking performance by estimating how the target object, as well as the other objects in the scene, move over time.

2.2.1 Tracking with Optical Flow

Recently, some trackers [11, 33, 40] use optical flow estimation to improve performance on instance tracking. FlowTrack [40] uses flow to warp features from previous frames to improve the feature representation and tracking accuracy. The warped feature maps are weighted by a spatial-temporal attention module; these feature maps are then input into subsequent correlation filter layers along with feature maps of the current frame. Other work [11] uses optical flow to obtain deep motion features, and then fuses appearance information with deep motion features for visual tracking. For hand-crafted features, deep image features, and deep motion features, the method separately learns a filter by minimizing the SRDCF [4] objective and then averages the filter responses to get final confidence scores. SINT+ [33] uses flow to remove motion inconsistent candidates. Specifically, it uses the estimated optical flow to map the locations of the pixels covered by the predicted box in the previous frame to the current frame, and remove the candidate boxes which contain less than 25% of those pixels.

However, none of these methods use a segmentation mask or flow uncertainty for tracking; our experiments demonstrate that both of these components are crucial for optimal tracking performance. We develop a probabilistic framework to use flow uncertainty for tracking.

2.2.2 Tracking with Uncertainty

There have been several recent works on estimating confidence in optical flow [17, 20, 26]. FlowNetH [17] is shown to be able to generate effective uncertainty estimates without the need of sampling or ensembles. As far as we know, these methods have not been used to improve performance of instance tracking. We propose a new framework which combines flow uncertainty estimates with appearance scores from a one-shot-detection method; we show that our method can significantly improve tracking robustness and obtains state-of-the-art tracking results.

Chapter 3

Background

3.1 Siamese Networks for Tracking

Our method is based on the SiamMask [35] framework, which is the state-of-the-art method on the VOT tracking benchmarks [21, 22, 23, 24]. It consists of siamese subnetwork for feature extraction and a region proposal subnetwork for bounding box proposal generation. The framework scores proposals based on an appearance matching score d, a size change penalty p_s, and a position change penalty p_c. The size change penalty p_s penalizes changes to the size of the bounding box of size w by h from one frame to the next; this is defined as

$$p_s = e^{(1-\max(\frac{r}{r'},\frac{r'}{r})\cdot\max(\frac{s}{s'},\frac{s'}{s}))\cdot k_p} \tag{3.1}$$

where s and s' are the padded areas p of the proposal box and the bounding box in the previous frame, respectively, given by

$$s^2 = (w + p) \times (h + p) \tag{3.2}$$

and r and r' are the aspect ratios of the proposal box and the bounding box in the previous frame, respectively. The score f for each proposal is calculated as

$$f = (1 - k_c) \cdot p_s \cdot d + k_c \cdot p_c \tag{3.3}$$

where k_c and k_p are hyperparameters. The position penalty p_c is obtained by penalizing the position of the center of the bounding box according to one period of a cosine function, centered at the position of the previous bounding box; the period of this cosine penalty is determined based on the size of the previous bounding box. The appearance matching score d is given by the output of the one-shot detection network, which matches a template image of the target object to the scene.

SiamMask [35] first chooses a bounding box based on the proposal with the highest score f. For the highest scoring proposal box, it then predicts a mask F_t, thresholds it into a binary mask $\hat{F}_t$ using threshold t_{seg}, and outputs the minimum bounding rectangle (MBR) of the binary mask as the final prediction of the location of the tracked object.

Chapter 4

Method

We introduce a new method which improves tracking robustness under distractors and large appearance changes. We visualize our pipeline in Figure 4.1. Our method uses optical flow to estimate the probability of being in part of the foreground for every pixels in a frame, which we call it a "FlowMask." Based on this probability mask, we assign a flow score for each proposal, which we combine with an appearance score to obtain the final tracking output. The rest of this section explains how our method works in detail.

4.1 Flow Mask

Our method makes use of previous work for uncertainty-aware dense optical flow estimation [17] that computes the probability that each pixel i in frame t corresponds to a given pixel location j in frame $t-1$: $p(c(I_{i,t}) = I_{j,t-1})$ where c maps pixel $I_{i,t}$ to a pixel in frame $t-1$. Given images I_t and I_{t-1}, this method predicts the probability of each pixel being part of the foreground as a Laplace distributions, parametrized by flow mean μ and scale b:

$$p(c(I_{i,t}) = I_{j,t-1}) = \mathcal{L}(I_{j,t-1} - I_{i,t}|\mu, b) \tag{4.1}$$

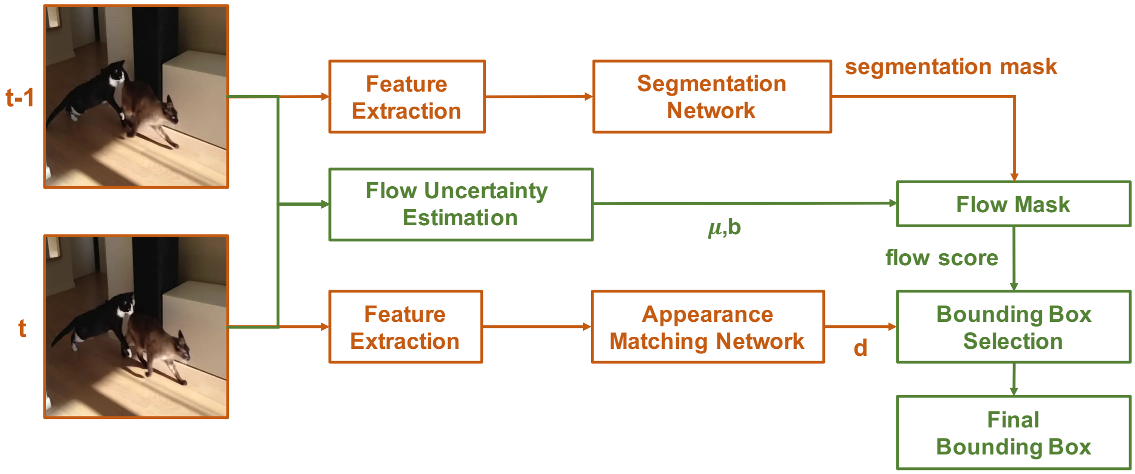

Figure 4.1: Overall pipeline of our method: On top of SiamMask [35](showed in orange), we add a module(showed in green) to compute flow score for each proposal using flow uncertainty estimations and segmentation output from previous frame; our method then combines flow scores with appearance scores to choose a bounding box proposal.

where the Laplace distributions are defined in the standard manner as

$$\mathcal{L}(u|\mu, b) = \frac{1}{2b} \exp\left(\frac{-|u-\mu|}{b}\right) \tag{4.2}$$

For notational convenience, we are omitting the conditioning for these probabilities on the images I_t and I_{t-1}. As we will show, flow uncertainty is crucial for robust tracking. Using Eqn. 4.1, we can compute the probability that pixel $I_{i,t}$ corresponds to pixel $I_{j,t-1}$. We compute the probability that pixel $I_{j,t-1}$ belongs to the target object. To do so, we use a segmentation-based tracking method [35] to obtain a "segmentation mask", which gives the probability that each pixel $I_{j,t-1}$ belongs to the foreground of the previous bounding box (i.e. the tracked object): $p(I_{j,t-1} \in F_{t-1})$ where F_{t-1} is the set of foreground pixels, i.e. the set of pixels in frame $t-1$ that belong to the tracked object. We combine these flow probabilities with the Segmentation Mask probabilities to estimate the probability that a pixel $I_{i,t}$ in frame t belongs to the

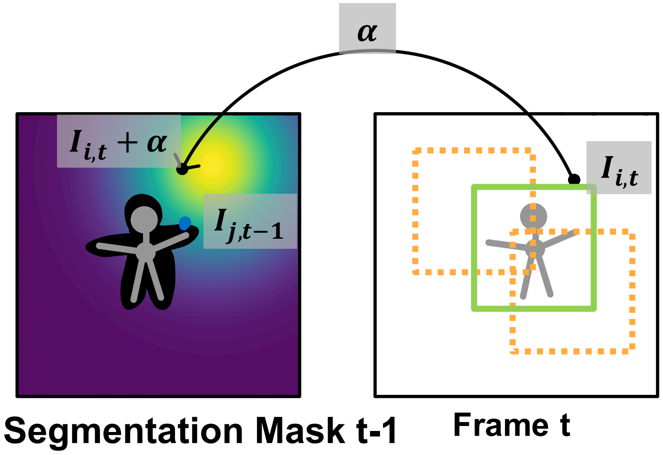

Figure 4.2: Illustration of how flow mask is computed. The green box is the ground-truth box; the orange boxes are proposals; α_u and α_v are the predicted flow mean from frame t to frame $t-1$ in u, v direction; colormap in frame $t-1$ visualizes a Laplace distribution parametrized by predicted flow mean and variance. The blue dot represents a point $I_{j,t-1}$ that belongs to the foreground in frame $t-1$.

tracked object:

$$p(I_{i,t} \in F_t) = \sum_j p(c(I_{i,t}) = I_{j,t-1})\, p(I_{j,t-1} \in F_{t-1}) \tag{4.3}$$

We compute the foreground probability for every pixel $I_{i,t}$ in frame t; we refer to the resulting set of probabilities as the "FlowMask" at frame t. This computation is implemented in a differentiable manner, and could be used in end-to-end trainable pipelines. This idea is further illustrated in Figure 4.2.

4.2 Flow Score

After we compute the flow mask at frame t, we can compute a flow score for each proposal $box_{(i,t)}$, denote as $f_s(box_{(i,t)})$ by averaging the foreground probabilities for each pixel in the box:

$$f_s(box_{(i,t)}) = \frac{1}{N_{box(i,t)}} \sum_{I_{i,t} \in box_{(i,t)}} p(I_{i,t} \in F_t)$$

where $N_{box(i,t)}$ represents the total number of pixels in $box_{(i,t)}$. However, one discrepancy in this score is that, even though $box_{(i,t)}$ is a rectangle, the object being tracked may not be shaped as a rectangle, which could cause $f_s(box_{(i,t)})$ to be much less than 1. This variability in $f_s(box_{(i,t)})$ will make it difficult to combine the flow score with the appearance score, as described in Section 4.3 below. To deal with this issue, we first compute the number of pixels that are in the thresholded segmentation mask $\hat{F}_{t-1}$ as $N_{\hat{F}_{t-1}}$. Then we compute a $t_{\text{flow},t}$ by dividing $N_{\hat{F}_{t-1}}$ by the number of pixels in previous frame's axis-aligned detection box box^*_{t-1}.

$$t_{\text{flow},t} = \frac{N_{\hat{F}_{t-1}}}{N_{box^*_{t-1}}} \tag{4.4}$$

This results in a flow score defined as

$$f'_s(box_{(i,t)}) = \min\left(\frac{f_s(box_{(i,t)})}{t_{\text{flow},t}}, 1\right) \tag{4.5}$$

4.3 Bounding Box Selection

Lastly we combine our flow score with the appearance score from the one-shot detection framework to obtain a motion score for a given proposal box:

$$(1 - k_f) \cdot p_c + k_f \times f'_s \tag{4.6}$$

where k_f is a hyperparameter, p_c was described in Section 3.1, and f'_s is obtained from Eqn. 4.5. We combine our position penalty p_c from Eqn. 4.6 with the size penalty p_s and appearance matching score d in Eqn. 3.3 to obtain the total score for each proposal box. Our entire pipeline is end-to-end differentiable, so we could backprop through our network and learn the value for the different hyperparameters. However, since there are only three hyperparameters, we proceed as in SiamMask [35] to do a hyperparameter searches and find the proposal box with the highest score to obtain the tracking output and estimate a segmentation mask for this box.

Chapter 5

Results

5.1 Implementation Details

Our method uses the pretrained SiamMask [35] network to obtain the appearance matching score d and to compute the segmentation mask. Since SiamMask [35] reports its tracking performace on visual object tracking datasets (VOT 2016, 2018 and 2019), we also report our performance on these three datasets. In SiamMask [35], they perform hyperparameter searches on $k_c \in [0.40, 0.43]$ and $k_p \in [0, 1]$, and we similarly search in these ranges; we also perform a random hyperparameter search for $k_f \in [0, 1]$ for Eqn. 4.6.

5.2 Quantitative Results

5.2.1 Evaluation for VOT

This section includes results on VOT2016 [21], VOT2018 [23], and VOT2019 [24]. VOT2016 consists of 60 video sequences. The VOT2017 [22] challenges replaces the 10 least challenging sequences with new ones. VOT2018 contains the same 60 video sequences as in VOT2017. VOT2019 replaces 20% of the videos in VOT2018 with new ones. The performance is evaluated in terms of accuracy (average overlap while tracking successfully), robustness (failure times), and Expected Average Overlap (EAO), which takes account of both accuracy and robustness, as is common for the

	VOT2016			VOT2018			VOT2019		
	EAO ↑	R ↓	A ↑	EAO ↑	R ↓	A ↑	EAO ↑	R ↓	A ↑
ATP [25]	-	-	-	-	-	-	0.394	0.291	0.650
Our Method	0.47	0.196	0.647	0.41	0.234	0.605	0.306	0.426	0.599
SiamMask [35]	0.433	0.214	0.639	0.38	0.276	0.609	0.283	0.467	0.596
UInet [25]	-	-	-	-	-	-	0.254	0.468	0.561
SiamMsST [25]	-	-	-	-	-	-	0.252	0.552	0.575
MemDTC [37]	0.297	1.310	0.5297	0.2651	1.5287	0.4909	0.252	0.552	0.575
CSRDCF [30]	0.338	0.85	0.51	-	-	-	0.201	0.632	0.496

Table 5.1: Results on VOT 2016, VOT2018, and VOT2019. R represents robustness and A represents accuracy. The top three performing trackers are colored with red and green respectively.

VOT challenges.

The results are shown in Table 5.1. We compare our method with all the other state-of-the-art trackers that predict both a bounding box for tracking and a segmentation mask for each frame. As can be seen, our method significantly improves over most state-of-the-art baselines in all categories across VOT2016, 2017, and 2018. In particular, our method builds upon [35], so the improvement should be judged relative to this method. However, our method for incorporating flow uncertainty into tracking is modular and can be combined with other state-of-the-art tracking methods as well.

In terms of speed, our methods operates at 5.62 frames per second, or 178ms per frame, including 18ms for SiamMask [35] and 60ms for the optical flow computation [17].

5.2.2 Ablations

Our method builds upon SiamMask [35] and incorporates flow uncertainty based on the previous frame's predicted segmentation mask to improve performance. To further investigate the importance of different components of our method, as well as the effectiveness of different approximations that we make, we conduct the following ablations:

Importance of Optical Flow We first investigate the importance of using optical flow for tracking, rather than the approach taken by several recent papers of using

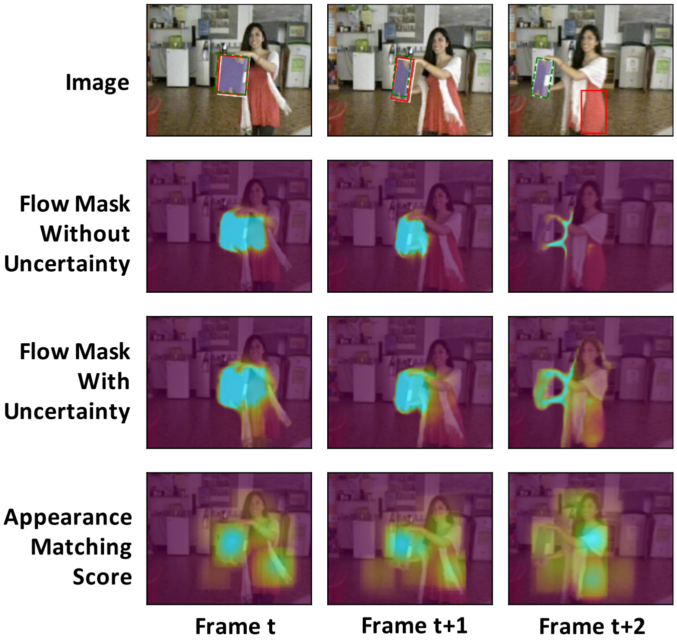
Image
Flow Mask Without Uncertainty
Flow Mask With Uncertainty
Appearance Matching Score
Frame t
Frame t+1
Frame t+2

Figure 5.1: Illustration of importance of uncertainty: One example that flow mask with uncertainty successfully tracks the target object but flow mask without uncertainty fails. In this figure, white boxes represent ground-truth boxes, red boxes represent prediction by using flow mask without uncertainty, green boxes shows the prediction using flow mask with uncertainty.

a cosine [2, 27, 35, 38] or Gaussian penalty [28] to penalize large motions from the previous frame. To analyze this, we note that our method for optical flow uses a Laplacian distribution, as shown in Equations 4.1 and 4.2. Thus, to evaluate the importance of optical flow, we replace the estimated flow distribution with a constant Laplacian, with 0 mean $\mu = (0, 0)$ and fixed scale parameters b. This ablation is referred to as "Ours minus Flow" (Ours - Flow) in Table 5.2. As can be seen, using a constant Laplacian distribution (rather than optical flow) leads to no improvement over the baseline SiamMask [35].

Importance of Optical Flow with Uncertainty For the next ablation, we probe the importance of utilizing uncertainty estimates for optical flow in tracking. To evaluate this, we fix the Laplacian scale parameters b to a constant. The result is shown in Table 5.2 as "Ours - Uncertainty." Although the result is better than our baseline, it still has a large performance gap with our method. As a qualitative analysis, Figure 5.1 shows one example where our method successfully tracks the target object but Ours-Uncertainty fails due to errors in the flow estimate under large object motion and perspective changes. This shows how the uncertainty increases the tracking robustness.

Importance of Segmentation Mask In this ablation, we investigate the importance of using a segmentation mask for the computation of the flow mask. In our method, we use SiamMask [35] to predict a segmentation mask for the previous frame. We then combine the probability of being in the foreground with the flow probability, as shown in Equation 4.3. We probe the importance of having a segmentation mask by replacing it with a bounding box. We analyze using both an axis-aligned bounding box (ALB) and a minimum bounding rectangle (MBR) mask (see SiamMask [35] for details). The result is shown in Table 5.2 as "Ours - SegMask (ALB)" and "Ours - SegMask (MBR)". As we can see, using the minimum bounding rectangle instead of a segmentation mask results in no improvement over the baseline. On the other hand, using the mask obtained using axis-align box improve upon baseline but still is not as effective as our method.

5.2.3 SiamMask+Flow Rejection

Last, we compare to an additional baseline that also uses optical flow to improve tracking. Following SINT+ [33], we evaluate using optical flow to filter out motion inconsistent candidates, and try to use this "flow rejection" method to improve the SiamMask [35]. Specifically, we use flow to warp the pixels covered by the predicted box in the previous frame. We then remove all proposals in the current frame that contain less than 25% of the warped pixels (this is similar to the procedure from SINT+ [33]). We refer to this experiment as "SiamMask plus flow rejection" (SiamMask + FlowRej) in Table 5.2. As we can see, using flow rejection does not improve the performance compared to the baseline SiamMask [35]. This degradation in performance, especially in robustness, is likely due to occasional errors in the flow estimation. This supports our claim about the important of flow uncertainty estimation for robust tracking.

	VOT2018		
	EAO↑	Robustness↓	Accuracy↑
Ours	**0.41**	**0.234**	0.605
Ours - Flow	0.38	0.276	0.609
Ours - Uncertainty	0.383	0.262	0.610
Ours - SegMask (ALB)	0.388	0.267	**0.614**
Ours - SegMask (MBR)	0.372	0.253	0.593
SiamMask [35] + FlowRej	0.361	0.290	0.613
SiamMask [35]	0.38	0.276	0.609

Table 5.2: Ablation Analysis. Ours-Flow uses identity flow; Ours-Uncertainty uses fixed variance; Ours-SegMask (ALB) replaces segmentation mask with an axis-aligned bounding box; Ours-SegMask (MBR) replaces segmentation mask with a minimum bounding rectangle.

5.3 Qualitative Analysis

Our method effectively improves tracking robustness under distractors and large object appearance changes. To better illustrate the effect of our method, we analyze our results qualitatively. In Figure 1.1, we visualize three cases where the state-of-the-art tracker SiamMask [35] fails but our method is able to successfully keep track of the target objects. For each case, we visualize the position penalty that

SiamMask [35] uses, the appearance matching score (i.e appearance_score) produced by the one-shot-detection network, and the flow mask introduced in this work in Section 4.1. Figure 1.1 shows two categories of challenging tracking scenarios:

Distractors One type of common failure case occurs when there are distractor objects in the background that are similar in appearance or category to the object being tracked. An example is shown in Figure 1.1(b), in which the target object runs across another person in the background. In this case, the appearance matching score (from the one-shot-detection network of SiamMask) is high for both people. The position penalty is also not useful in this case due to the fast motion of the target object. Thus, if we only rely on the appearance matching score and the position penalty, we would track the distractor instead of the target object, as illustrated by the red detection box (output by SiamMask).

Nevertheless, the flow mask successfully tracks the target object. Since the flow mask is a probabilistic estimate based on the predicted segmentation mask of the previous frame, it is able to focus precisely on the target object; additionally, because we incorporate flow uncertainty, our method is also robust to small errors in the estimated flow.

Large Appearance Changes Another challenging problem in tracking is large appearance changes. In Figure 1.1(a), the image of the target object becomes blurry under large camera motion. In this scenario, the appearance matching network predicts similar confidence for many areas in the image. The position penalty also fails because the position of the large change in the object position on the image due to the fast camera motion. Similarly, in Figure 1.1(c), the position penalty is also not effective due to the fast motion of the target object. In this case, the deformation of the target object (a bird) also causes the appearance matching score to be uncertain, leading to a failure from SiamMask.

However, in both cases, our proposed flow mask is still able to track the target object. These examples illustrate that our method is robust to large appearance changes, such as blurry images and deformations, as well as fast moving objects.

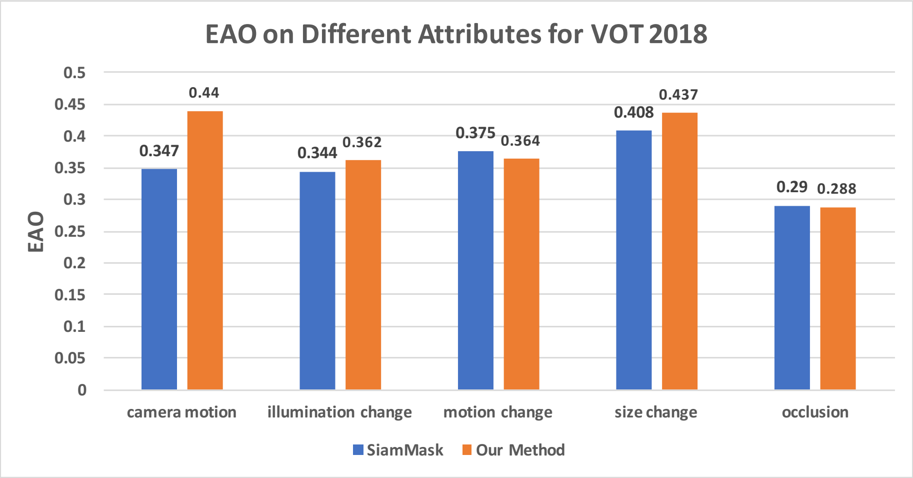

Figure 5.2: Results breakdown on VOT 2018 for different visual attributes. We compare the EAO under these five attributes of our method with the baseline SiamMask [35].

5.4 Detailed Analysis on VOT2018

To better understand the effect of using our method, we perform in-depth analysis on the VOT 2018 dataset. In the VOT 2018 dataset, each frame is manually labeled with five visual attributes that reflect a particular challenge: (i) camera motion, (ii) motion change, (iii) size change, (iv) illumination change, (v) occlusion. In case that a frame doesn't correspond to any of those five attributes, it is labeled as ”non-degraded”. Those labels enable us to analyze the benefits of our method while focusing only on the frames that contain a given attribute.

The results are shown in Figure 5.2, in which we compare our method to the SiamMask [35] baseline that we build on top of. As can be seen, our method significantly improves the tracker's performance under camera motion, and we also see modest improvements under size change and illumination change. Our method performance slightly worse than SiamMask [35] under motion change and occlusion.

In Figure 5.3, we visualize one example of a failure case due to occlusion. In this case, the target object gets occluded by another similar looking distractor. We find that, when an occlusion occurs, the predicted segmentation mask tends to also mask the distractor; thus it will mislead the calculation of FlowMask in the following frames.

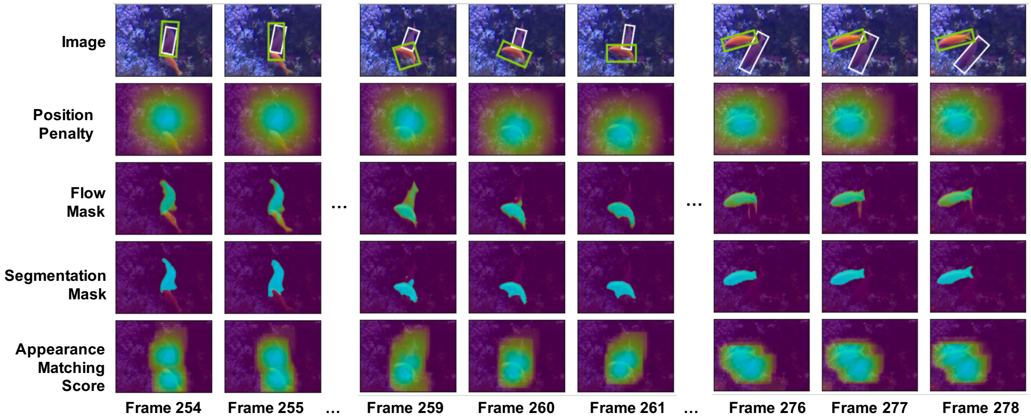

Figure 5.3: Illustration of a failure case due to occlusion: when there is an occlusion, the predicted segmentation mask and flow mask tends to drift to the distractor. In this figure, white boxes represent the ground-truth , green boxes show the prediction from our method.

Eventually both FlowMask and the segmentation mask would have high confidence for the distractor. Thus tracker would drift and track the distractor instead. In SiamMask [35] baseline, it fails similarly in this case. However, since the location of two objects don't change too much, eventually baseline method recovers by the help of position penalty. In our case, the use of flow mask prevent us from recovering from a failure in this case.

Chapter 6

Conclusions

In this paper, we introduce a novel probabilistic framework that combines appearance and flow uncertainty for tracking. We show that our method, when evaluated on Visual Object Tracking datasets, significantly improves the performance of a state-of-the-art tracker. Ablation experiments show the importance of each component of our framework, such as the use of flow uncertainty and warping a segmentation mask. We hope that our work can be insightful to future research on robust tracking under distractor objects and large object appearance changes.

Bibliography

[1] Tamer Başar. A new approach to linear filtering and prediction problems. 2001. 2.1

[2] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip H S Torr. Fully-convolutional siamese networks for object tracking. In *ECCV 2016 Workshops*, pages 850–865, 2016. 1, 2.1, 5.2.2

[3] David Bolme, J. Beveridge, Bruce Draper, and Yui Lui. Visual object tracking using adaptive correlation filters. pages 2544–2550, 06 2010. 2.1

[4] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4310–4318, 2015. 2.2.1

[5] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4310–4318, 2015. 2.1

[6] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016. 2.1

[7] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, 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. 2.2

[8] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3057–3065, 2017. 2.1

[9] H. K. Galoogahi, T. Sim, and S. Lucey. Multi-channel correlation filters. In *2013 IEEE International Conference on Computer Vision*, pages 3072–3079, Dec 2013. 2.1

[10] Hamed Kiani Galoogahi, Terence Sim, and Simon Lucey. Correlation filters with limited boundaries. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4630–4638, 2014. 2.1

[11] Susanna Gladh, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Deep motion features for visual tracking. *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1243–1248, 2016. 2.2.1

[12] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *ECCV*, 2016. 2.1

[13] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:583–596, 2015. 2.1

[14] B.K. Horn and B.G. Schunck. Determining optical flow. In *Artificial Intelligence*, 1981. 2.2

[15] K. Hossain and Chi-Woo Lee. Visual object tracking using particle filter. *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 98–102, 2012. 2.1

[16] 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. 2.2

[17] Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1, 2.2.2, 4.1, 5.2.1

[18] Michael Isard and Andrew Blake. Condensation—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998. 2.1

[19] Dae-Sik Jang, Seok-Woo Jang, and Hyung-Il Choi. 2d human body tracking with structural kalman filter. *Pattern Recognition*, 35:2041–2049, 2002. 2.1

[20] Claudia Kondermann, Daniel Kondermann, Bernd Jähne, and Christoph S. Garbe. An adaptive confidence measure for optical flows based on linear subspace projections. In *DAGM-Symposium*, 2007. 2.2.2

[21] Matej Kristan, Aleš Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin Zajc, Tomas Vojir, Gustav Häger, Alan Lukežič, and Gustavo Fernandez. The visual object tracking vot2016 challenge results. Springer, Oct 2016. 3.1, 5.2.1

[22] Matej Kristan, Aleš Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin Zajc, Tomas Vojir, Gustav Häger, Alan Lukežič, Abdelrahman

Eldesokey, and Gustavo Fernandez. The visual object tracking vot2017 challenge results, 2017. 3.1, 5.2.1

[23] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, Gustavo Fernandez, and et al. The sixth visual object tracking vot2018 challenge results, 2018. 3.1, 5.2.1

[24] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kamarainen, Luka Čehovin Zajc, Ondrej Drbohlav, Alan Lukezic, Amanda Berg, Abdelrahman Eldesokey, Jani Kapyla, and Gustavo Fernandez. The seventh visual object tracking vot2019 challenge results, 2019. 3.1, 5.2.1

[25] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kamarainen, Luka Čehovin Zajc, Ondrej Drbohlav, Alan Lukezic, Amanda Berg, Abdelrahman Eldesokey, Jani Kapyla, and Gustavo Fernandez. The seventh visual object tracking vot2019 challenge results, 2019. **??**, **??**, **??**

[26] Jan Kybic and Claudia Nieuwenhuis. Bootstrap optical flow confidence and uncertainty measure. *Computer Vision and Image Understanding*, 115:1449–1462, 2011. 2.2.2

[27] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 1, 2.1, 5.2.2

[28] Feng Li, Xiaohe Wu, Wangmeng Zuo, David Zhang, and Lei Zhang. Remove cosine window from correlation filter-based visual trackers: When and how. *ArXiv*, abs/1905.06648, 2019. 2.1, 5.2.2

[29] Xutang Li, Shanzhen Lan, Yue Jiang, and Pin Xu. Visual tracking based on adaptive background modeling and improved particle filter. *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pages 469–473, 2016. 2.1

[30] Alan Lukezic, Tomás Vojír, Luka Cehovin Zajc, Juan E. Sala Matas, and Matej Kristan. Discriminative correlation filter tracker with channel and spatial reliability. *International Journal of Computer Vision*, 126:671–688, 2017. **??**

[31] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3074–3082, 2015. 2.1

[32] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1164–1172, June 2015. 2.2

[33] Ran Tao, Efstratios Gavves, and Arnold W. M. Smeulders. Siamese instance search for tracking. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1420–1429, 2016. 2.2.1, 5.2.3

[34] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J. Black. Video segmentation via object flow. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2.2

[35] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr. Fast online object tracking and segmentation: A unifying approach. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1328–1338, June 2019. (document), 1, 1.1, 1, 2.1, 3.1, 3.1, 4.1, 4.1, 4.3, 5.1, **??**, 5.2.1, 5.2.2, 5.2.3, **??**, **??**, 5.3, 5.2, 5.4

[36] Shiuh-Ku Weng, Chung Ming Kuo, and Shu-Kang Tu. Video object tracking using adaptive kalman filter. *J. Visual Communication and Image Representation*, 17:1190–1208, 2006. 2.1

[37] T. Yang and A. B. Chan. Visual tracking via dynamic memory networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. **??**

[38] Y. Zha, M. Wu, Z. Qiu, S. Dong, F. Yang, and P. Zhang. Distractor-aware visual tracking by online siamese network. *IEEE Access*, 7:89777–89788, 2019. ISSN 2169-3536. 1, 1, 2.1, 5.2.2

[39] 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. 2.2

[40] Zheng Zhu, Wei Wu, Wei Zou, and Junjie Yan. End-to-end flow correlation tracking with spatial-temporal attention. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2.2.1