Towards Robust and Accurate Visual Object Tracking: Scale Estimation

A DISSERTATION PRESENTED
BY
HAOYI MA
TO
THE FACULTY OF SCHOOL OF ENGINEERING AND APPLIED SCIENCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF
ELECTRICAL AND COMPUTER ENGINEERING

UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA
MAY 2022

©2022 – Haoyi Ma
all rights reserved.

Thesis advisors: Professor Zongli Lin and Professor Scott T. Acton Haoyi Ma

Towards Robust and Accurate Visual Object Tracking: Scale Estimation

Abstract

Visual object tracking is a critical task with a plethora of applications ranging from medical imaging to robotics. In a video sequence, given the states of the target object in the initial frame, the goal of visual object tracking is to estimate the target state in the subsequent frames. The tracking problem is challenging as numerous factors affect the performance of a tracking algorithm, such as scaling, viewpoint, illumination variation and occlusion. Furthermore, as a basic building block of many time-critical systems, another major challenge is that a visual tracker should meet the strict constraints of the time and computational budgets.

Recently, visual object tracking has seen advances made possible by the development of the tracking-by-detection paradigm, in which the tracking problem is formulated as a classification problem typically addressed by a machine learning technique used to discriminate the target appearance from the surrounding background. With the advancement of discriminative modeling and feature representations, many important works have been carried out, such as the discriminative correlation filter based trackers and Siamese network based trackers. However, most tracking methods cannot accommodate large scale variations in complex video sequences and thus result in unsatisfactory performance.

In this thesis, we focus on providing robust and accurate scale estimation methods to further enable accurate and robust visual object tracking. First, a novel criterion, the average peak-to-correlation energy, is incorporated into a multi-scale searching framework to obtain robust scale estimation. The resulting system is referred to as *SITUP: Scale Invariant Tracking Using average Peak-to-correlation energy*.

Second, to address the problem of the heavy computational load associated with the multi-scale searching scheme in SITUP, we investigate different strategies to reduce the computational cost. The resulting system is named *FAST: Fast and Accurate Scale estimation for Tracking*. In comparison with SITUP, FAST obtains comparable performance while operating at a frame rate up to three times higher.

Third, to further enable aspect ratio adaptability, an effective scale estimation scheme is proposed to adapt to aspect ratio variation via using a group of discriminative correlation filters to localize the target boundary, and the resulting system is named *TARA: Tracking with Aspect Ratio Adaptability*.

Last, to obviate the need of hyper-parameters associated with the candidate boxes (*e.g.*, scale factors of the multi-scale searching scheme, or sizes and aspect ratios of the predefined candidate anchor boxes), we formulate tracking as parallel classification and regression problems, and the target object is directly classified and the corresponding bounding boxes are regressed in a unified fully convolutional network. The resulting system is named *CAT: Centerness-aware Anchor-free Tracker*. Extensive experiments on publicly available tracking benchmark datasets show that our proposed scale estimation methods, with their appealing features, achieve salient performance in a wide variety of tracking scenarios.

Index Terms: Visual object tracking, scale estimation, aspect ratio adaptability, discriminative correlation filter, Siamese network, fully convolutional network, anchor-free, centerness.

Contents

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Objectives and Contributions . . . 5
- 1.3 Dissertation Outline . . . 8

2 SITUP: Scale Invariant Tracking using Average Peak-to-Correlation Energy 10
- 2.1 Background . . . 11
- 2.2 Related Work . . . 13
- 2.3 Proposed Method . . . 17
- 2.4 Experiments . . . 26
- 2.5 Discussion . . . 34

3 FAST: Fast and Accurate Scale Estimation for Tracking 35
- 3.1 Background . . . 36
- 3.2 Proposed Method . . . 37
- 3.3 Experiments . . . 42
- 3.4 Discussion . . . 47

4 TARA: Tracking with Aspect Ratio Adaptability 49
- 4.1 Background . . . 50
- 4.2 Proposed Method . . . 52
- 4.3 Experiments . . . 57
- 4.4 Discussion . . . 66

5 CAT: Centerness-Aware Anchor-Free Tracker 68
- 5.1 Background . . . 70
- 5.2 Related Work . . . 71
- 5.3 Proposed Method . . . 73
- 5.4 Experiments . . . 81
- 5.5 Discussion . . . 91

6 Conclusions and Future Work **92**
6.1 Conclusions 92
6.2 Future Work 93
6.3 Publications 94

References **103**

Listing of figures

1.1 An illustration of a discriminative tracker 2
1.2 The tracking pipeline of a classical discriminative correlation filter based tracker . . 3
1.3 The tracking pipeline of a classical Siamese network based tracker 4

2.1 Qualitative comparisons between SITUP and two representative scale adaptive discriminative correlation filter based trackers 16
2.2 A comparison between the average peak-to-correlation energy measure and the naïve maximum response value 22
2.3 A qualitative comparison between tracker SITUP and tracker SITUP_naïve . . . 23
2.4 The tracking pipeline of SITUP 24
2.5 Quantitative attribute-based comparisons between SITUP and SITUP_naïve . . . 28
2.6 The precision and success plots for the overall performance of SITUP and two representative scale adaptive discriminative correlation filter based trackers 31
2.7 The precision and success plots of SITUP and two representative scale adaptive discriminative correlation filter based trackers on the scale variation attribute 33

3.1 The tracking pipeline of FAST 40
3.2 The precision and success plots for the overall performance of SITUP and FAST . 44
3.3 The precision and success plots of SITUP and FAST on the scale variation attribute 46

4.1 A failure case of SITUP facing the aspect ratio variation challenge 51
4.2 The comparison between the state representation employed in TARA and the standard state representation 54
4.3 The tracking pipeline of TARA 56
4.4 The precision and success plots for the overall performance of TARA and three representative scale adaptive discriminative correlation filter based trackers 60
4.5 Qualitative comparisons between TARA and three representative scale adaptive discriminative correlation filter based trackers 61
4.6 The precision and success plots of TARA and three representative scale adaptive discriminative correlation filter based trackers on the scale variation attribute 63

4.7 The precision and success plots of TARA and three representative scale adaptive discriminative correlation filter based trackers on the aspect ratio variation attribute . 65

5.1 The tracking pipeline of CAT 74
5.2 Qualitative comparisons between CAT and two representative scale adaptive Siamese network based trackers 83
5.3 The precision and success plots for the overall performance of CAT and two representative scale adaptive Siamese network based trackers 85
5.4 The precision and success plots of CAT and two representative scale adaptive Siamese network based trackers on the scale variation attribute 87
5.5 The precision and success plots of CAT and two representative scale adaptive Siamese network based trackers on the aspect ratio variation attribute 89

List of Tables

2.1 Comparisons between SITUP and two scale adaptive discriminative correlation filter based trackers in terms of area-under-curve score and precision score 30
2.2 Comparisons between SITUP and two scale adaptive discriminative correlation filter based trackers regarding area-under-curve score and precision score on the scale variation attribute 32

3.1 Comparisons between FAST and SITUP in terms of area-under-curve score, precision score and speed 43
3.2 Comparisons between FAST and SITUP on the scale variation attribute in terms of area-under-curve score and precision score 45
3.3 Comparisons between FAST and its three variants regarding area-under-curve score, precision score and speed 47

4.1 Comparisons between TARA and three discriminative correlation filter based trackers in terms of area-under-curve score, precision score, and speed 59
4.2 Comparisons between TARA and three discriminative correlation filter based trackers on the scale variation attribute in terms of area-under-curve and precision scores 62
4.3 Comparisons between TARA and three scale adaptive discriminative correlation filter based trackers on the aspect ratio variation attribute regarding area-under-curve score and precision score. 64
4.4 Comparisons between TARA and its two variants regarding area-under-curve score and precision score. 66

5.1 Comparisons between CAT and two scale adaptive Siamese network based trackers in terms of area-under-curve score, precision score and speed 84
5.2 Comparisons between CAT and two scale adaptive Siamese network based trackers on the scale variation attribute regarding area-under-curve and precision scores . . 86
5.3 Comparisons between CAT and two scale adaptive Siamese network based trackers on the aspect ratio variation attribute regarding area-under-curve and precision scores 88
5.4 Comparisons between CAT and its three variants in terms of area-under-curve score and precision score 90

To my parents, Guang Ma and Fengqiu Han
To my fiancée, Jiayue
To my cats, Maylee and Bolt
For their kind blessings, accompany, and care

Acknowledgments

It feels so surreal when I really start writing this dissertation to present to you what I have done in the past six years. I am so blessed to have the support from my best advisors, my mentor, my friends and my beloved families along the path. I cannot make this far without all of your help.

First, I would like to thank my advisors Prof. Zongli Lin and Prof. Scott T. Acton. I would like to thank Prof. Lin for always guiding me with his patience, and continuous support along my whole PhD journey. I learned so much from him about the theory of life and how to be a researcher. Also, I would like to show my appreciation to Prof. Acton, who opened the door of image processing to me. His insightful feedback always pushes me to sharpen my thinking and brings my research to a higher level. I am always inspired by his energy and how he keeps his humor in the daily life. Prof. Acton once joked with me that my first paper writing sounds like "high school", I truly hope this dissertation sounds like "graduate school".

Also, I would like to thank Prof. Nikos Sidiropoulos for mentoring my teaching and showing me how to improve the presentation skill. During my first class as a lecturer, he encouraged me and said "As long as your derivatives are positive, you will be good", which always encourages me to keep moving when I am feeling down.

Many thanks to my committee members, Prof. Gang Tao, Prof. Tom Fletcher, and Prof. Lin Ma, I appreciate all the help and every suggestion from them. They are so kind and encouraging, and I am really honored to have them on my committee to share my work with.

I am so grateful to be part of the VIVA Lab and Prof. Lin's Lab, where I gained so much knowledge and friendship. I will always treasure the precious memories with Dr. Jie Wang, Dr. Tiffany Ly, Dr. Nazia Tabassum, Dr. Narsin Sadeghzadehyazdi, Dr. Tamal Batabyal, Dr. Andrea Vaccari, Soumee Guha, Tanjin Taher Toma, Samarth Singh, Dr. Matthew Korban, Dr. Yusheng Wei, Dr. Shize Su, Dr. Syed Ali Asad Rizvi, Dr. Yijing Xie, Dr. Weiqi Bai, Tingyang Meng, Qilong Song, and Matthew Carson.

I would like to send my gratitude to the VITAL lab, Prof. Miaomiao Zhang's Lab and Prof. Tom Fletcher's lab. Special thanks go to Jian Wang, Jiarui Xing, and Haoyi Liang, it is always a pleasure to discuss research with them.

Last, I would like to thank my parents, my beloved fiancee and my cats for their unconditional love, kindness, and caring. You are my inspiration and my energy. This dissertation is for you!

1 Introduction

1.1 Background

Visual object tracking (VOT) is an important problem in image processing with applications ranging from video surveillance to medical imaging [1–9]. Given the initial state of a target object in an initial frame, the generic goal of tracking is to estimate the state of the target object in the following frames. This problem is challenging since the arbitrary target object is solely defined by its initial

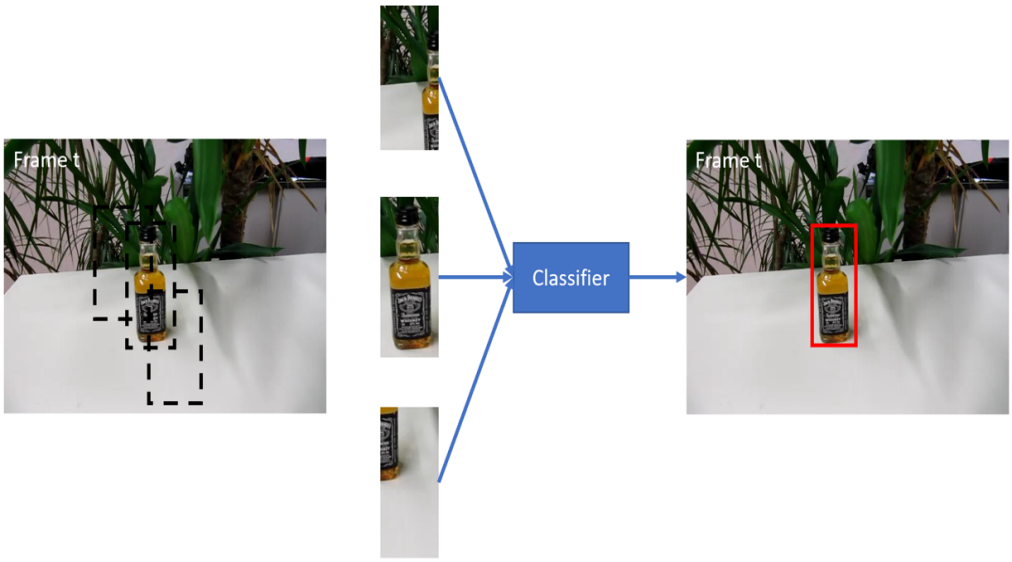

Figure 1.1: An illustration of a discriminative tracker. To track the target object, the candidate patches (marked using black dashed lines) are sampled around the estimated target object location in the previous frame. Then, the corresponding feature representations are extracted and provided to the classifier, and the candidate patch with the highest classification score is selected as the tracking result for the current frame (marked using red solid lines).

state, and numerous complicated interfering factors affect the performance of a tracking algorithm, such as viewpoint change, illumination variation, shape deformation, and occlusion. Besides, since VOT is a basic building block of many time-critical systems, another major challenge is that a visual tracker should meet the strict constraints of time and computational budget, especially with respect to mobile operating systems or embedded computing architectures where real-time analysis is often desired and resources are limited.

In general, VOT methods can be divided into two different groups: *generative* and *discriminative*. Generative tracking methods [10–13] learn a model to represent the appearance of a target object, and then the tracking problem is formulated as finding the object appearance most similar to the model. As shown in Figure 1.1, instead of building a model to describe the appearance of an object, discriminative tracking methods [14–16] aim to discriminate the target object from the background via

training a classifier. Recently, with the advancement of discriminative modeling and feature representations, discriminative tracking methods are often found to outperform generative methods in terms of accuracy. Discriminative trackers can usually run efficiently using inexpensive hand-crafted features [17,18] and various learning methods such as a structured output support vector machine [14], multi-expert entropy minimization [15], and discriminative correlation filters [16,19,20].

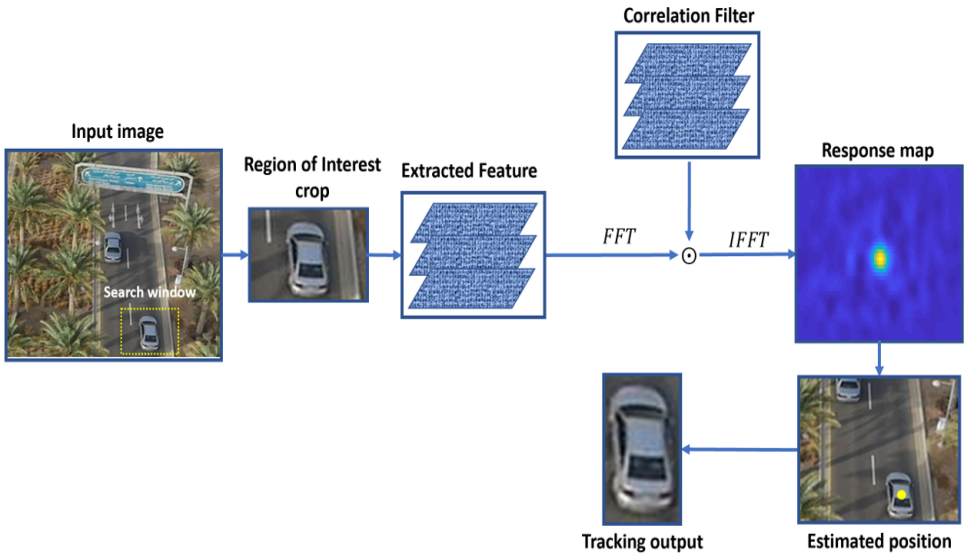

Figure 1.2: The tracking pipeline of a classical discriminative correlation filter based tracker. The features corresponding to the region of interest (marked using yellow dashed lines) are computed and transformed into the frequency domain via the fast Fourier transformation (FFT). Then, the response map in the time domain is obtained by applying the inverse fast Fourier transformation (IFFT) to the product of the discriminative correlation filter and the extracted features. The position of the maximum response value of the response map is treated as the center position of the target object in the current frame.

Saliently, among the discriminative tracking methods, discriminative correlation filters (DCF) [16,19] (to be described in Section 2.3.1) and deep Siamese network [21] (to be described in Section 5.3.1) have been the two most prominent paradigms for VOT. In DCF based trackers, the dense sampling scheme is approximated by generating a circulant matrix, of which each row denotes a circular shift of the base sample. An illustration of the tracking pipeline of a classical DCF based tracker is pre-

sented in Figure 1.2. By exploiting the circular correlation, the correlation filter can be computed efficiently in the Fourier domain using only fast Fourier transforms (FFTs) and pointwise operations. In such a case, the regression model can be computed in the Fourier domain, which brings significant speed improvement in both training and testing processes.

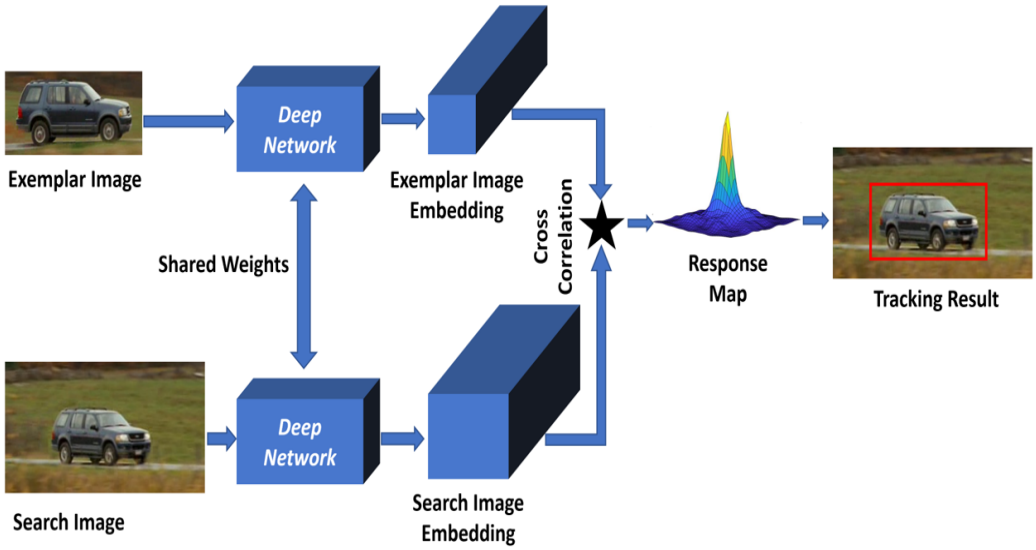

Figure 1.3: The tracking pipeline of a classical Siamese network based tracker. The exemplar image and search image are embedded into the same feature space, and the response map can be computed using the cross correlation operation.

In Siamese network based trackers, the tracking problem is formulated as a similarity learning problem, and the network is trained to locate the target object within the search image. As depicted in Figure 1.3, the network takes an exemplar image and a search image as input. The image patch with the center point as the target center is cropped in the initial frame and then adopted as the exemplar image. At each frame, an image patch centered at the previous estimated target object location is cropped and treated as the search image. Then, the Siamese network applies an identical transformation to both the exemplar image and the search image, and the similarity score is computed via the cross-correlation operation.

In recent years, the tracking community has been largely focusing on discriminative correlation filters and deep Siamese networks since these two paradigms have significantly improved the tracking performance. However, in the standard setting of these two paradigms, the output state only includes the vertical and horizontal coordinates of the target object center in a video frame. Scale variations of the target object often occur due to changes in the target object appearance or motion along the camera axis, and failing to adapt to the scale variation will yield inferior tracking performance. Besides, scale estimation plays an important role in many applications, such as single cell tracking[22], in which scale estimation can provide important biological information. Therefore, achieving accurate scale estimation of the target object is beneficial to tracking in many aspects by providing a more accurate tracking result. Robust and accurate scale estimation is challenging and is further complicated by the presence of other challenging factors, such as motion blur and occlusion. Currently, most existing tracking methods cannot accommodate large scale variations in complex image sequences, and thus yield inferior tracking performance.

1.2 Objectives and Contributions

This thesis aims to provide accurate and robust scale estimation methods for VOT under different tracking scenarios (*e.g.*, with only limited computing resources and with advanced GPUs) to boost the tracking performance. In order to achieve this goal, we propose multiple scale estimation methods. Our contributions are summarized as follows.

- **Contribution 1: scale invariant tracking using average peak-to-correlation energy**
  The standard DCF based tracker is restricted to translation estimation since it is a template based method with a fixed template size. The standard DCF based tracker will yield inferior tracking performance when a large scale variation of the target object is encountered. Within the existing scale estimation methods, the criterion employed to obtain the scale estimation

is the naïve maximum response value. In contrast, we argue that the robustness of the maximum response value will be heavily degraded due to the presence of some other challenging factors such as motion blur and occlusion. We propose to incorporate a novel criterion, the average peak-to-correlation energy (APCE), into the multi-scale searching scheme to obtain robust and accurate scale estimation via exploiting the robustness of the APCE measure. The concise integration of the APCE measure in the multi-scale searching scheme makes our scale estimation method generic and straightforward to be incorporated into any DCF based tracker. The resulting system is named SITUP: scale invariant tracking using average peak-to-correlation energy. To the best of our knowledge, SITUP is the first work to address the scale estimation problem from the perspective of the robustness of the criterion. By robustness of the criterion, we refer to the capability of better representing the confidence level of the tracking result under challenging scenarios (*e.g.,* motion blur and occlusion). Extensive experiments show that SITUP effectively tackles the problem of fixed template size in the standard discriminative correlation filter based tracker while operating in real-time on a single CPU.

- **Contribution 2: fast and accurate scale estimation**

  The multi-scale searching scheme employed in SITUP is computationally expensive when facing large scale variations. To address the associated computational cost, we investigate different strategies to yield higher computational efficiency. Besides, the simple linear interpolation scheme employed in SITUP for model updating is proved to be sub-optimal in reference 23, since only the current frame is taken into account while updating the model, and the contribution of previous frames in the model fades out exponentially. Differently from SITUP, we employ a more robust updating scheme presented in reference 23, which considers the current frame and all previous frames simultaneously. The resulting system is

named FAST: fast and accurate scale estimation for tracking. Comparing to SITUP, FAST obtains comparable performance while operating at a three times higher frame rate, which validates the effectiveness of the employed computational cost reduction strategy.

- **Contribution 3: tracking with aspect ratio adaptability**

  Comprehensive evaluations using notable benchmark datasets demonstrate that our proposed scale searching frameworks (SITUP and FAST) can handle large scale variation challenges effectively. However, the multi-scale searching framework employed in SITUP and FAST is not born with aspect ratio adaptability. Besides, the employed multi-scale searching framework is not flexible enough due to the predefined scale sampling manner. We propose to address the scale estimation problem and enable aspect ratio adaptability by utilizing a group of discriminative correlation filters to localize the boundaries of the target object. Furthermore, rich deep hierarchical convolutional features of deep convolutional neural networks that encode the target object appearance information with different levels of abstraction are exploited for robust visual visual tracking. The resulting system is named TARA: tracking with aspect ratio adaptability. Extensive empirical evaluation using publicly available tracking benchmark datasets demonstrates that TARA can meet the demand of scale and aspect ratio variation challenges.

- **Contribution 4: centerness-aware anchor-free tracker**

  Visual trackers based on the Siamese network have received significant attention due to their balanced accuracy and speed. In the pioneering work [21], a fully convolutional Siamese network (SiamFC) is achieved via training a Siamese network to perform template matching and compute a similarity score between the exemplar image of the target object and the search image. In order to incorporate scale estimation into the Siamese network based tracker, most scale estimation methods rely either on a multi-scale searching scheme or on refining a set

of predefined anchor boxes. To achieve satisfactory performance, heuristically selected parameters, such as scale factors of the multi-scale searching scheme, or sizes and aspect ratios of the predefined candidate anchor boxes, entail tedious and heuristic tuning. On the contrary, by exploiting the expressive power of the fully convolutional network, we design a centerness-aware anchor-free tracker (CAT), where the location and scale of the target object are predicted in an anchor-free fashion by decomposing tracking into parallel classification and regression problems. The proposed anchor-free design obviates the need for hyperparameters (*e.g.,* sizes, aspect ratios, and numbers) related to the anchor boxes, making CAT more generic and flexible. Within the proposed tracker, with the observation that many low-quality predictions are produced corresponding to the locations far from the target center. We design a centerness-aware classification branch to identify the foreground from the background while predicting the normalized distance from the location within the foreground to the target center, *i.e.*, the centerness. The proposed centerness-aware classification branch improves the tracking accuracy and robustness significantly by suppressing low-quality state estimates. Extensive experiments show that the proposed tracker CAT obtains achieves salient performance in a wide variety of tracking scenarios.

1.3 Dissertation Outline

In Chapter 2, we first review the tracking problem and the related scale estimation problem. Then, we introduce related scale adaptive variants of the standard DCF based tracker and our proposed scale adaptive DCF based tracker SITUP. In Chapter 3, the strategies employed in FAST for computational cost reduction are described in detail. In Chapter 4, the aspect ratio variation challenges and related DCF based trackers are reviewed, and our proposed scale searching scheme with aspect ratio adaptability TARA is detailed. In Chapter 5, we first introduce Siamese network based trackers and

anchor-free mechanism that are closely related to our method. Then, we introduce CAT, our proposed Siamese network based tracker, in detail. In the last chapter, we conclude the dissertation and point out some potential future work.

2

SITUP: Scale Invariant Tracking using Average Peak-to-Correlation Energy

The objective of this chapter is to introduce a scale adaptive discriminative correlation filter (DCF) based tracker, named SITUP, in which a novel criterion called the average peak-to-correlation energy (APCE) is incorporated into the multi-resolution translation filter framework to obtain robust and accurate scale estimation. In most existing scale estimation methods, the criterion employed to

obtain the scale estimation is the naïve maximum response value. With the observation that robustness of the naïve maximum response value will heavily degrade in challenging scenarios, we propose to incorporate the APCE measure into the multi-resolution translation filter framework. Through exploiting the robustness of the APCE measure, SITUP tackles the scale estimation problem effectively, and provides superior performance over other scale adaptive variants of standard DCF based trackers while operating in real-time on a single CPU. Besides, the concise integration of the APCE measure in an exhaustive scale searching framework makes our scale estimation method generic and easy to be incorporated into any DCF based tracker.

The rest of this chapter is organized as follows. Section 2.2 gives an overview of the prior works most relevant to our proposed approach. The proposed method is detailed in Section 2.3. Specifically, Section 2.3.1 details the multi-channel kernelized correlation filter, which we adopt as the baseline tracker. In Section 2.3.2, details about the employed features are given. Our scale searching strategy is described in Section 2.3.3. The details of the experiment are presented in Section 2.4. Specifically, in Section 2.4.1, the implementation details are presented. The benchmark datasets and evaluation protocols are described in Section 2.4.2. The comparison results of the performed experiments are presented in Section 2.4.3 and Section 2.4.4. Finally, Section 2.5 concludes the chapter.

2.1 Background

It can be said that motion is the *raison d'être* for vision. Without motion, organisms simply would not require a vision system. At the core of emulating the biological ability to track objects in motion is the task of visual object tracking (VOT). The core problem of VOT is: given an initial state of a target object in an initial frame, compute the state of the target object in the remainder of the video frames. Often, tracking systems are foiled by complicating factors such as illumination variation, partial and full occlusion.

A key signal processing technology for tracking is found in the DCF. DCF has been successfully applied to VOT characterized by both high accuracy and high efficiency. By exploiting the circular correlation, the correlation filter can be computed efficiently in the Fourier domain using only fast Fourier transforms (FFTs) and pointwise operations [16,19]. However, the standard DCF based tracker employs a fixed size template and is therefore performance limited when scale variation is encountered.

To address the scale estimation problem, several approaches have been proposed in references 24–27. The discriminative scale space tracker (DSST) [24] tackles the scale estimation problem by learning two separate correlation filters for explicit translation and scale estimation. One straightforward approach is the scale adaptive tracker with multiple features (SAMF) [25], which tackles the scale estimation problem by applying a standard two-dimensional correlation filter at multiple resolutions. Another approach is the joint scale spatial filter [26], which incorporates scale estimation by proposing a three-dimensional correlation filter in the joint spatial and scale space. This estimation is achieved by modeling the exhaustive template searching in the joint space as a block-circulant matrix. An adjustable Gaussian window function and a keypoint based model are proposed in reference 27 to tackle the scale estimation problem.

In all the aforementioned scale estimation methods, the criterion employed to obtain the scale estimation is the naïve maximum response value. In contrast, we argue that the robustness of the maximum response value will be heavily degraded due to the presence of some other challenging factors such as motion blur, partial and full occlusion. Here, in SITUP, instead of using the naïve maximum response value to obtain both the translation and scale estimation as in the aforementioned methods, we propose to use the APCE measure as the criterion to obtain the scale estimation. The exhaustive scale searching strategy is employed in our tracker. First, at the training stage, a two-dimensional correlation filter is trained in the standard DCF setting. At the testing stage, we obtain the sample patches centered around the estimated target location of the previous frame with

different scales and resize them into a fixed size. The correlation filter response maps are obtained by comparing the resized sample patches with the learned model. The APCE measure of the response map at each scale is computed and the response map with the largest APCE measure is selected as the one with the best scale estimation. The translation estimation is obtained by searching for the position of the maximum response value within the response map with the largest APCE measure. After obtaining the translation and scale estimation, we update the learned model using the interpolation update strategy.

To the best of our knowledge, SITUP is the first work to address the scale estimation problem from the perspective of the robustness of the criterion. The concise integration of the APCE measure in an exhaustive scale searching framework makes the scale estimation method generic and easily employed in other DCF based trackers to further boost their tracking performance. Besides, our method benefits from a reliable update of the target appearance model, as the APCE measure can reflect the confidence level of the response map.

2.2 Related Work

VOT is one of the fundamental problems in image processing. Here, we consider the single object tracking task where the target object is identified in the first video frame, and is to be tracked in the following frames. This problem is challenging since the arbitrary target object is defined solely by its initial location and scale.

Typically, VOT methods work by building a target appearance model from the observed image information by way of a generative [10–12] or a discriminative model [16,19,20]. Generative appearance models describe the target appearance with statistical models or templates. Discriminative tracking methods, instead, implement machine learning techniques to discriminate the target appearance from the surrounding background by formulating VOT as a classification problem.

Saliently, DCF based methods [16,19,20] have been successfully applied to VOT and have achieved high efficiency and high accuracy. These methods have shown to provide excellent results on tracking benchmark datasets [28,29], while maintaining real-time speed. The DCF based trackers locate the target object in the new frame by learning a discriminative correlation filter to discriminate the target object from the background. Bolme *et al*. [16] propose to train the correlation filter via minimizing the total squared error between the actual and the desired correlation outputs on a set of grayscale sample patches. By utilizing circular correlation, the authors show that the resulting correlation can be computed efficiently in the Fourier domain, which brings hundreds of time speed improvement in both the training and testing stages. Henriques *et al*. [19,20] further show that the DCF framework can be equivalently formulated as a regression problem on the set of all cyclic shifts of the training sample patches.

To generalize the DCF framework to include multi-dimensional features, several solutions have been proposed that attempt to learn an exact multi-channel filter from the set of the training samples. However, computational cost has plagued such approaches. To remediate, approximate formulations for learning multi-channel filters [19,23] have been proposed for VOT that have proven to be robust while scaling linearly with the number of channels.

In many different tracking scenarios, the DCF based approaches have demonstrated the capability of locating the target accurately. However, the standard DCF based tracker is restricted to translation estimation since it is a template based method with a fixed template size. The standard DCF based tracker will yield inferior performance when a large scale variation of the target object is encountered. Hence, achieving accurate scale estimation of the target object is beneficial to tracking in many aspects by providing a more accurate tracking result.

Danelljan *et al*. [24] propose DSST method that, by directly learning the appearance changes induced by scale variation, trains two separate correlation filters for translation estimation and scale estimation, respectively. A two-dimensional standard discriminative correlation filter is employed

for translation estimation and another one-dimensional discriminative correlation filter is employed for scale estimation. However, the unreliable translation estimation might lead to inferior scale estimation in DSST, which results in a poor generalization capability. Alternatively, Li and Zhu [25] propose the SAMF method to tackle the scale estimation problem by extending a standard DCF based tracker for translation estimation to multiple resolutions, and the scale and translation estimation are obtained jointly. SAMF is incorporated into many DCF based tracking methods due to its good generalization capability. However, SAMF suffers from inferior performance compared to the state-of-the-art trackers.

SAMF is the solution most closely related to our proposed approach, since we also employ a multi-resolution translation filter framework. Here, instead of using the naïve maximum response value as in reference 25, we propose to employ the APCE measure as the criterion to obtain the best scale estimation first and the translation estimation is further obtained. Due to the robustness of the APCE measure in different scenarios, as shown in Figure 2.1, our approach obtains superior performance while maintaining a real-time speed despite its simplicity.

Figure 2.1: Qualitative comparisons between SITUP and two representative scale adaptive DCF based trackers. In comparison with other scale adaptive variants of standard DCF based trackers, including scale adaptive tracker with multiple features (SAMF), and discriminative scale space tracking (DSST), our approach significantly increases both the robustness and accuracy by estimating the target object size accurately. Example frames shown are from *Human3* (top row), *Lemming* (middle row) and *Dudek* (bottom row) sequences that are selected from online tracking benchmark datasets to involve the scale variation challenge. The bounding boxes in the first column identify the target objects to be tracked in corresponding sequences, and the number on the upper-left corner of each image is the frame number of corresponding image.

2.3 Proposed Method

In Section 2.3.1 and Section 2.3.2, the formulation of our baseline tracker multi-channel kernelized correlation filter and the employed features are described in detail, respectively. The scale searching strategy is described in detail in Section 2.3.3.

2.3.1 The Kernelized Correlation Filter Tracker

SITUP leverages the kernelized correlation filter (KCF) tracker [19]. By exploiting the structure of the circulant matrix with high efficiency, the discriminative ability of the KCF tracker is enhanced with the augmentation of negative samples while maintaining high speed. In the formulation of KCF, the generated data matrix has a circular structure by using the cyclic shifts of the base sample to approximate the dense samples over the base sample. For notational simplicity, we start with one-dimensional signals. Given one dimensional data $\mathbf{x} = [v_1, v_2, ..., v_n]^{\mathrm{T}}$. A cyclic shift of $\mathbf{x}$ is denoted as

$$\mathbf{P}\mathbf{x} = [v_n, v_1, v_2, ..., v_{n-1}]^{\mathrm{T}}, \tag{2.1}$$

where $\mathbf{P}$ is a permutation matrix defined as

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}_{n \times n} . \tag{2.2}$$

The full set of the cyclic shifts of base sample can be written as

$$\{\mathbf{P}^u\mathbf{x}|u = 0, 1, ..., n-1\}, \tag{2.3}$$

which can further be concatenated to generate the data matrix

$$\mathbf{X} = C(\mathbf{x}) = \begin{bmatrix} v_1 & v_2 & v_3 & \dots & v_n \\ v_n & v_1 & v_2 & \dots & v_{n-1} \\ v_{n-1} & v_n & v_1 & \dots & v_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_2 & v_3 & v_4 & \dots & v_1 \end{bmatrix}_{n\times n}. \tag{2.4}$$

Matrix $C(\mathbf{x})$ is called a circulant matrix since the matrix is purely generated by the cyclic shifts of $\mathbf{x}$. One intriguing property of the circulant matrix is that all circulant matrices can be diagonalized by the discrete Fourier transform (DFT) [30], regardless of the generating vector $\mathbf{x}$. This can be expressed as

$$\mathbf{X} = \mathbf{F}\text{diag}(\hat{\mathbf{x}})\mathbf{F}^{\text{H}}, \tag{2.5}$$

where $\mathbf{F}$ is the DFT matrix that is independent of $\mathbf{x}$, $\mathbf{F}^{\text{H}}$ is the Hermitian transpose of $\mathbf{F}$, *i.e.*, $\mathbf{F}^{\text{H}} = (\mathbf{F}^*)^{\text{T}}$, and $\mathbf{F}^*$ is the complex-conjugate of $\mathbf{F}$. $\hat{\mathbf{x}}$ denotes the DFT of the generating vector $\mathbf{x}$, *i.e.*, $\hat{\mathbf{x}} = \mathcal{F}(\mathbf{x})$.

In DCF, the goal of training is to find $f(\mathbf{z}) = \mathbf{w}^{\text{T}}\mathbf{z}$ that minimizes the squared error over samples $\mathbf{x}_i$ and their regression targets y_i as follows:

$$\min_{\mathbf{w}} \sum_{i}^{n} (f(\mathbf{x}_i) - y_i)^2 + \lambda\|\mathbf{w}\|^2. \tag{2.6}$$

The scalar λ is a regularization parameter that controls overfitting. This ridge regression problem has

a closed-form solution, which can be written as

$$\mathbf{w} = (\mathbf{X}^{\mathrm{H}}\mathbf{X} + \lambda)^{-1}\mathbf{X}^{\mathrm{H}}\mathbf{y}. \tag{2.7}$$

Each row of the data matrix $\mathbf{X}$ corresponds to one sample $\mathbf{x}_i$, each element of $\mathbf{y}$ is a regression target y_i. The decomposition of the circulant matrix can be utilized to simplify the solution of the regression problem. Substitution of Equation (2.5) in Equation (2.7) results in the solution,

$$\hat{\mathbf{w}}^* = \frac{\hat{\mathbf{x}}^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda}, \tag{2.8}$$

where $\hat{\mathbf{x}}^*$ is the complex-conjugate of $\hat{\mathbf{x}}$. This solution reduces the computational cost of both extracting patches explicitly and solving a general regression problem, since it only involves the DFT and element-wise operation.

To allow for a more powerful classifier with nonlinear regression functions $f(\mathbf{z})$, the solution $\mathbf{w}$ is expressed as a linear combination of the samples

$$\mathbf{w} = \sum_i \alpha_i \phi(\mathbf{x}_i), \tag{2.9}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^{\mathrm{T}}$ is the variable under optimization in dual space, as opposed to the primal space $\mathbf{w}$. $\phi(\cdot)$ denotes the mapping to a nonlinear feature space. The kernel trick can be employed as

$$f(\mathbf{z}) = \mathbf{w}^{\mathrm{T}}\mathbf{z} = \sum_{i=1}^{n} \alpha_i \kappa(\mathbf{z}, \mathbf{x}_i). \tag{2.10}$$

For the most commonly used kernels (*e.g.*, Gaussian, linear and polynomial), the circulant matrix trick can also be employed, since these kernels treat each dimension of the data equally [19]. The dual

space coefficients $\boldsymbol{\alpha}$ can be computed as

$$\hat{\boldsymbol{\alpha}}^* = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{\mathbf{xx}} + \lambda}, \tag{2.11}$$

where $\mathbf{k}^{\mathbf{xx}}$ is the kernel correlation as defined in reference 19. In SITUP, the linear kernel is adopted in consideration of both speed and accuracy as

$$\mathbf{k}^{\mathbf{xx}'} = \mathcal{F}^{-1}(\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}}'). \tag{2.12}$$

Since the algorithm only requires element-wise operation and DFT/IDFT, the computational cost is at a nearly-linear $\mathcal{O}(n \log n)$. A large search window is employed to enclose more negative samples for training. Besides training, the circulant matrix trick can also be employed to speed up the detection process. In the next frame, the patch $\mathbf{z}$ at the same location is treated as the base sample and the corresponding response map in the Fourier domain is computed as

$$\hat{\mathbf{f}}(\mathbf{z}) = \hat{\mathbf{k}}^{\mathbf{xz}} \odot \hat{\boldsymbol{\alpha}}, \tag{2.13}$$

where $\mathbf{x}$ and $\mathbf{z}$ correspond to the base sample and the base patch, respectively. The response map in the spatial domain is computed through IDFT as

$$\mathbf{f}(\mathbf{z}) = \mathcal{F}^{-1}\left(\hat{\mathbf{f}}(\mathbf{z})\right). \tag{2.14}$$

When using IDFT to transform $\hat{\mathbf{f}}(\mathbf{z})$ back to the spatial domain, the translation with respect to the maximum response is treated as the translation of the target object. Since the template is of fixed size, the update process is straightforward. Two sets of coefficients should be updated, one being the dual space coefficients $\boldsymbol{\alpha}$, and the other the base sample $\mathbf{x}$. As the setting in reference 19, the

coefficients are updated through linear interpolation using

$$\mathbf{T} = \theta \mathbf{T}_{\text{new}} + (1 - \theta)\mathbf{T}, \tag{2.15}$$

where $\mathbf{T} = [\boldsymbol{\alpha}^{\mathrm{T}}, \mathbf{x}^{\mathrm{T}}]^{\mathrm{T}}$ is the template to be updated.

2.3.2 Multi-channel Feature Integration

Working in the dual space $\boldsymbol{\alpha}$ instead of the primal space $\mathbf{w}$ has the advantage of allowing multi-channel features. The kernel correlation function only needs to compute norms of the arguments. By summing the individual dot-product for each channel in the Fourier domain, a dot-product can be computed easily. Suppose that the multiple channels of the data representation are concatenated as $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2 ..., \mathbf{D}_c]$, the multi-channel extension of the linear kernel can be rewritten as

$$\mathbf{k}^{\mathbf{D}\mathbf{D}'} = \mathcal{F}^{-1}\left(\sum_c \hat{\mathbf{D}}_c^{*} \odot \hat{\mathbf{D}}_c^{'}\right). \tag{2.16}$$

Strong hand-crafted features rather than the grayscale pixel values can be employed in the tracker. Three types of features are utilized in the SITUP tracker including raw grayscale pixel values, histogram of oriented gradients, and color-naming.

The histogram of oriented gradients (HoG)[17] is effective for object detection and can be computed efficiently. The image is divided into small connected regions named cells and the gradient information of uniformly spaced cells is extracted. Then, the HoG counts occurrences of discrete gradient orientation to form the histogram.

Color names (CN) or color-naming are linguistic color labels, which are assigned by human to describe colors in the world. Recently, CNs have been adopted in different vision tasks such as visual object tracking[20] and action recognition[31], and have achieved promising results. A CN provides the

perception of the target object color, which contains salient information regarding the target object as shown in references 32 and 33.

Different features often correspond to different visual patterns; hence, some features will be more discriminative and robust than others in a particular circumstance. Here, we employ the discussed three features as complementary features to make our tracker more robust to different challenging scenarios.

2.3.3 Our Scale Searching Framework

Our scale searching scheme bears some similarity to the structure of SAMF, since we also employ a multi-resolution translation filter framework. In contrast to the SAMF approach, SITUP exploits the robustness of APCE criterion, which enables robust and accurate scale estimation when facing different scenarios. SAMF only employs a naïve maximum response value as the criterion, with which the associated robustness will be heavily degraded when scale variation presents with other challenging factors. Besides, our approach obtains a superior performance with a reliable update of the target appearance model since the APCE measure can better reflect the confidence level of the response map.

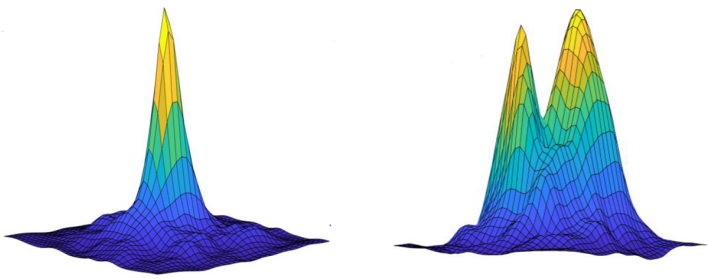

Figure 2.2: A comparison between the APCE measure and the naïve maximum response value. The left response map and the right response map have the same maximum response value, while the left response map has a higher APCE measure due to the sharper peak and less fluctuation.

The fluctuation and the peak value of the response map can reveal the confidence level of the tracking result. In the extreme case, the ideal response map should have only one sharp peak at the true target object location while being smooth and close to zero in all other areas when the detected target object is perfectly matched to the correct target object location and scale. A sharper peak will result in a higher localization accuracy. On the contrary, the response map will fluctuate fiercely, and the APCE measure will significantly decrease when challenged with severe occlusion and motion blur. Wang *et al.*[34] propose to employ the maximum response value and an APCE measure to provide a high-confidence update strategy for robustness. The APCE measure is defined as

$$APCE = \frac{|\mathbf{f}_{\max} - \mathbf{f}_{\min}|^2}{\text{mean}\left(\sum_{w,h}(\mathbf{f}_{w,h} - \mathbf{f}_{\min})^2\right)}, \tag{2.17}$$

where $\mathbf{f}_{\max}$, $\mathbf{f}_{\min}$, and $\mathbf{f}_{w,h}$ denote the maximum, the minimum and the w-th row h-th column elements of the response map $\mathbf{f}$, respectively.

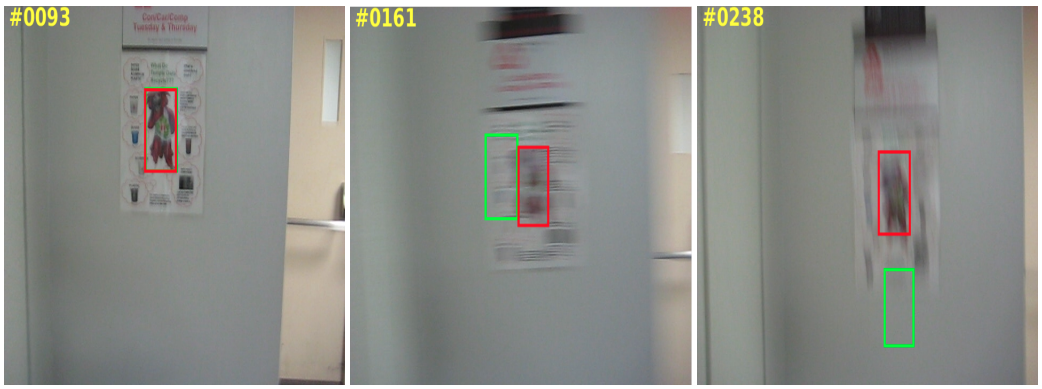

Figure 2.3: A comparison between tracker SITUP (in red) and tracker SITUP_naïve (in green). In comparison with SITUP_naïve that employs the maximum response value as the criterion within the same framework, benefiting from the robustness of the APCE measure, SITUP can accurately estimate the target size and location. In contrast, the robustness of the maximum response value will be heavily degraded due to other challenging factors (*e.g.*, motion blur) and SITUP_naïve cannot estimate the scale of the target object accurately. SITUP_naïve loses the target object due to inaccurate update of the target appearance model and erroneously selects a portion of the background. Example frames shown are from the *BlurOwl* sequence in the online tracking benchmark dataset, which involves four challenges with regard to scale variation, motion blur, fast motion and in-plane rotation.

The APCE criterion is more robust than the maximum response value criterion, which is noticeable when scale variation and other challenges such as motion blur and occlusion occur at the same time. As shown in Figure 2.3, the target object experiences four challenges of scale variation, motion blur, fast motion and in-plane rotation, and SITUP obtains accurate scale and translation estimation with the robust APCE measure. However, a tracker that employs the maximum response value as the criterion within the same framework cannot handle the scale variation due to the presence of other challenging factors. Hence, the tracker with the naïve maximum response value gets stuck in the background due to inaccurate update of the target appearance model and thus yields poor performance.

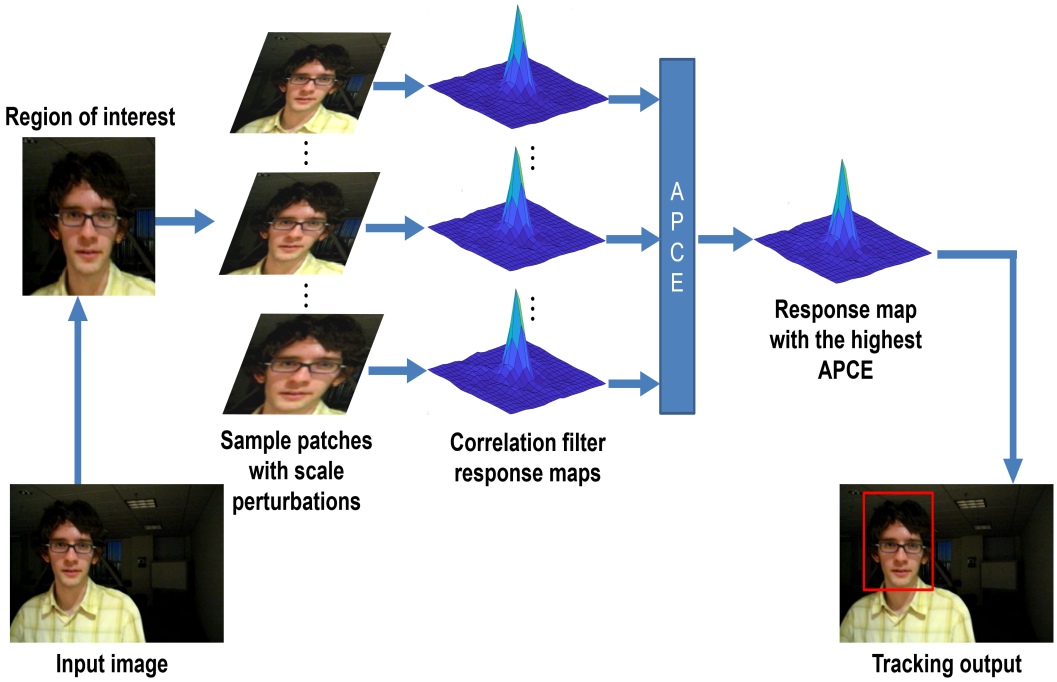

Figure 2.4: The tracking pipeline of SITUP. Patches with different scales are sampled centered around the previous estimated target object location and registered to the same size as the fixed template size. The response maps are computed respectively and the response map with the highest APCE measure is chosen as the one with the best scale estimation. The translation estimation is obtained by localizing the maximum response value within the response map with the highest APCE measure.

As shown in Figure 2.4, a standard two-dimensional discriminative correlation filter is trained first to provide preliminary translation estimation. Then, several patches centered around the estimated target location with different scale perturbations are sampled. All extracted patches are registered to the fixed template size. The response map of each patch is computed individually with the translation filter. The APCE measure of each response map is computed and the response map with the highest APCE measure is chosen as the one with the best scale estimation. The translation estimation is obtained by localizing the maximum response value within the response map with the highest APCE measure. With the new translation and scale estimation, the corresponding patch is extracted and employed to update the coefficient matrix and the target appearance model.

The template size is fixed as $\mathbf{s_T} = (s_x, s_y)$. The scaling pool is defined as $\mathbf{S} = \{t_1, t_2, ..., t_k\}$. Assume that the target searching window size of the last frame is $\mathbf{s}_{j-1}$. For the current jth frame, k patches with their sizes in $\{t_i \mathbf{s}_{j-1} | t_i \in \mathbf{S}\}$ are sampled and resized through bilinear-interpolation to the fixed template size $\mathbf{s_T}$. The response map of a resized patch is defined as $\mathbf{f}_{t_i}$ and computed using Equation (2.13) and Equation (2.14). The corresponding APCE measure is defined as APCE_{t_i} and is computed by Equation (2.17). The scale estimation is obtained by searching for the response map with the highest APCE measure as

$$t = \arg\max_{t_i}(\text{APCE}_{t_i}). \tag{2.18}$$

The translation estimation is obtained by searching for the location of the highest response value within the response map $\mathbf{f}_t$. Since the target movement is implied in the response map, the final displacement should be tuned by t to obtain the real translation bias. The update procedure will be implemented following Equation (2.15).

2.4 Experiments

Extensive evaluation is performed on three benchmark datasets to validate our approach. The implementation details are presented in Section 2.4.1. The benchmark datasets and evaluation protocols are described in Section 2.4.2. To demonstrate the robustness of the employed APCE measure, in Section 2.4.3, we present a comparison between SITUP and SITUP_naïve, which employs the naïve maximum response within the SITUP framework. A comparison between our approach and other scale adaptive variants of standard DCF based trackers are presented in Section 2.4.4.

2.4.1 Implementation Details

The regularization parameter is set as $\lambda = 0.0003$, and the learning rate is set as $\theta = 0.004$. The standard deviation of the desired correlation output is set to be $1/10$ of the target size. The padding ratio is set as 1.5 to make the search window 2.5 times the target size in terms of both width and height, which is typical in the existing literature.

For the scaling pool, we employ the same setting $\mathbf{S} = \{0.985, 0.99, 0.995, 1.0, 1.005, 1.01, 1.015\}$ as in SAMF[25]. The features we employ are computed by augmenting HoG, CN, and grayscale pixel values. To reduce computational cost, a 31-dimensional variant of HoG is employed as in references 19 and 25. The RGB space is transformed into a color-naming space, which is an 11-dimensional color representation. The grayscale features are normalized to the range $[-\frac{1}{2}, \frac{1}{2}]$. Each extracted feature channel of the sample patch is weighted by a cosine window to address the boundary effect. Also, we employ the linear kernel in consideration of both speed and accuracy in our tracker.

All tracking algorithms are implemented with Matlab on an Intel Core i7-7700K 4.5 GHz CPU with 16 GB RAM. A GeForce GTX 1080 GPU and MatConvNet 1.0-beta25[35] are used to reproduce the results of the trackers that require GPU implementation. Parameters of our tracking algorithm are fixed in all experiments. The parameters of other trackers are set according to the values in

their original code.

2.4.2 Datasets and Metrics

All trackers are quantitatively evaluated on the online tracking benchmark (OTB) datasets (OTB2013, OTB50 and OTB2015), following the one-pass evaluation (OPE) protocol described in references 28 and 29. OTB2013 is proposed in reference 28 for a comprehensive performance evaluation with 50 video sequences that is fully annotated with ground truth bounding boxes. To further address the problem of insufficient data, OTB2015 is proposed in reference 29 with 100 fully annotated video sequences. Since some target objects are similar or less challenging, 50 difficult and representative ones of OTB2015 are selected and represented as OTB50 for an in-depth analysis. The sequences in OTB datasets are categorized by way of 11 attributes. The 11 attributes are: illumination variation (IV), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-view (OV), background clutter (BC), and low resolution (LR).

To quantitatively evaluate the tracking results on OTB datasets, three standard evaluation metrics are exploited including precision score, success rate, and tracking speed in frames per second (fps). The precision is the relative number of frames in a video sequence for which the estimated location is within the specified threshold distance of the ground truth. The precision plot can be obtained by evaluating the precision score at different defined thresholds. The representative precision score (Prec.) for each tracker is reported at the threshold of 20 pixels as the setting in the standard OTB toolkit. The success rate is computed as the percentage of frames where the intersection-over-union between the tracked bounding box and the groundtruth bounding box exceeds the given threshold. The success plots show that the ratios of successful frames at the thresholds vary from 0 to 1. In terms of success plots, we use area-under-curve (AUC) scores to summarize the trackers.

2.4.3 The Robustness of the APCE Measure

To demonstrate the robustness of the APCE measure within SITUP in challenging scenarios, we compare the performance of SITUP and SITUP with the APCE measure replaced with the naïve maximum response value on OTB50, which contains 50 difficult and representative sequences. We denote SITUP with the naïve maximum response value as SITUP_naïve. The set of parameters in SITUP_naïve is the same as in SITUP to achieve a fair comparison.

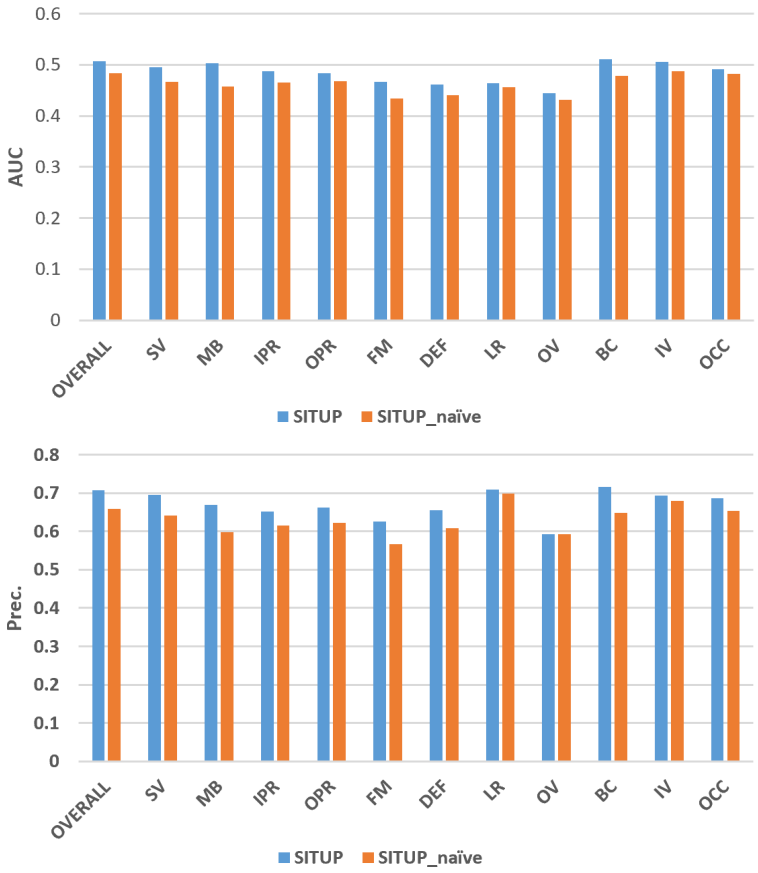

Figure 2.5: Specific attribute comparisons of SITUP and SITUP_naïve on OTB50 in terms of the area-under-curve (AUC) score and the precision score (Prec.) at a standard threshold of 20 pixels used in the literature. SITUP significantly outperforms SITUP_naïve in 8 out of 11 attributes and performs similarly in the remaining 3 attributes (low resolution (LR), out of view (OV) and illumination variation (IV)), which evidences the robustness of the APCE measure in challenging scenarios.

As shown in Figure 2.5, in comparison with SITUP_naïve, SITUP improves the AUC score by 2.3% and the precision score by 5.0%. For the attribute-based AUC score, SITUP outperforms SITUP_naïve by 2.8%, 4.6%, 2.3%, 1.5%, 3.3%, 2.1%, 0.8%, 1.2%, 3.3%, 1.8% and 1.0% on SV, MB, IPR, OPR, FM, DEF, LR, OV, BC, IV and OCC attributes, respectively. In term of the attribute-based precision score, SITUP outperforms SITUP_naïve by 5.4%, 7.2%, 3.7%, 4.1%, 5.8%, 5.7%, 1.0%, 0.1%, 6.7%, 1.4% and 3.4% on SV, MB, IPR, OPR, FM, DEF, LR, OV, BC, IV and OCC attributes, respectively.

In 8 out of 11 attributes (SV, MB, IPR, OPR, FM, DEF, BC and OCC), SITUP obtains promising improvement over SITUP_naïve. On other three attributes (LR, OV and IV), the performance of SITUP and SITUP_naïve are similar. Compared to the naïve maximum response value, the APCE measure is more robust in the most challenging scenarios and can provide a more accurate scale estimation within our scale searching framework, resulting in improved tracking performance.

2.4.4 Performance Comparison

Overall performance

To compare the overall performance of SITUP to other DCF based trackers, we execute three trackers on the OTB datasets (DSST, SAMF and our tracker SITUP). Table 2.1 summarizes the overall performance of the trackers in terms of the AUC score and the precision score for a threshold of 20 pixels (a typical parameter used in the literature), and the success plots and precision plots are given in Figure 2.6. As shown in Table 2.1 and Figure 2.6, the best results are obtained using SITUP.

Table 2.1: Comparisons with two scale adaptive variants of standard DCF based trackers on OTB2013, OTB50 and OTB2015 benchmark sequences. SITUP achieves the best performance in terms of the area-under-curve (AUC) score and the precision score at a standard threshold of 20 pixels used in the literature. The first and the second best values are highlighted in bold and underlined.

		OTB2013		OTB50		OTB2015	
Method	**Speed**	**Prec.**	**AUC**	**Prec.**	**AUC**	**Prec.**	**AUC**
DSST	23.43	0.734	0.550	0.620	0.457	0.682	0.513
SAMF	<u>27.19</u>	<u>0.772</u>	<u>0.572</u>	<u>0.656</u>	<u>0.472</u>	<u>0.747</u>	<u>0.551</u>
SITUP	**32.35**	**0.800**	**0.598**	**0.708**	**0.507**	**0.776**	**0.572**

In comparison with SAMF, which also employs a multi-resolution translation filter framework, SITUP improves the precision score by 2.8%, 5.2% and 2.9% and AUC score by 2.6%, 3.5% and 2.1% on OTB2013, OTB50 and OTB2015, respectively. It is worth mentioning that our approach achieves the largest performance improvement on OTB50, which contains the most difficult 50 sequences. As mentioned in Section 2.3.3, the employed criterion APCE of our tracker is more robust than the maximum response value especially in challenging tracking scenarios, which is evidenced by the comparison between SITUP and SAMF.

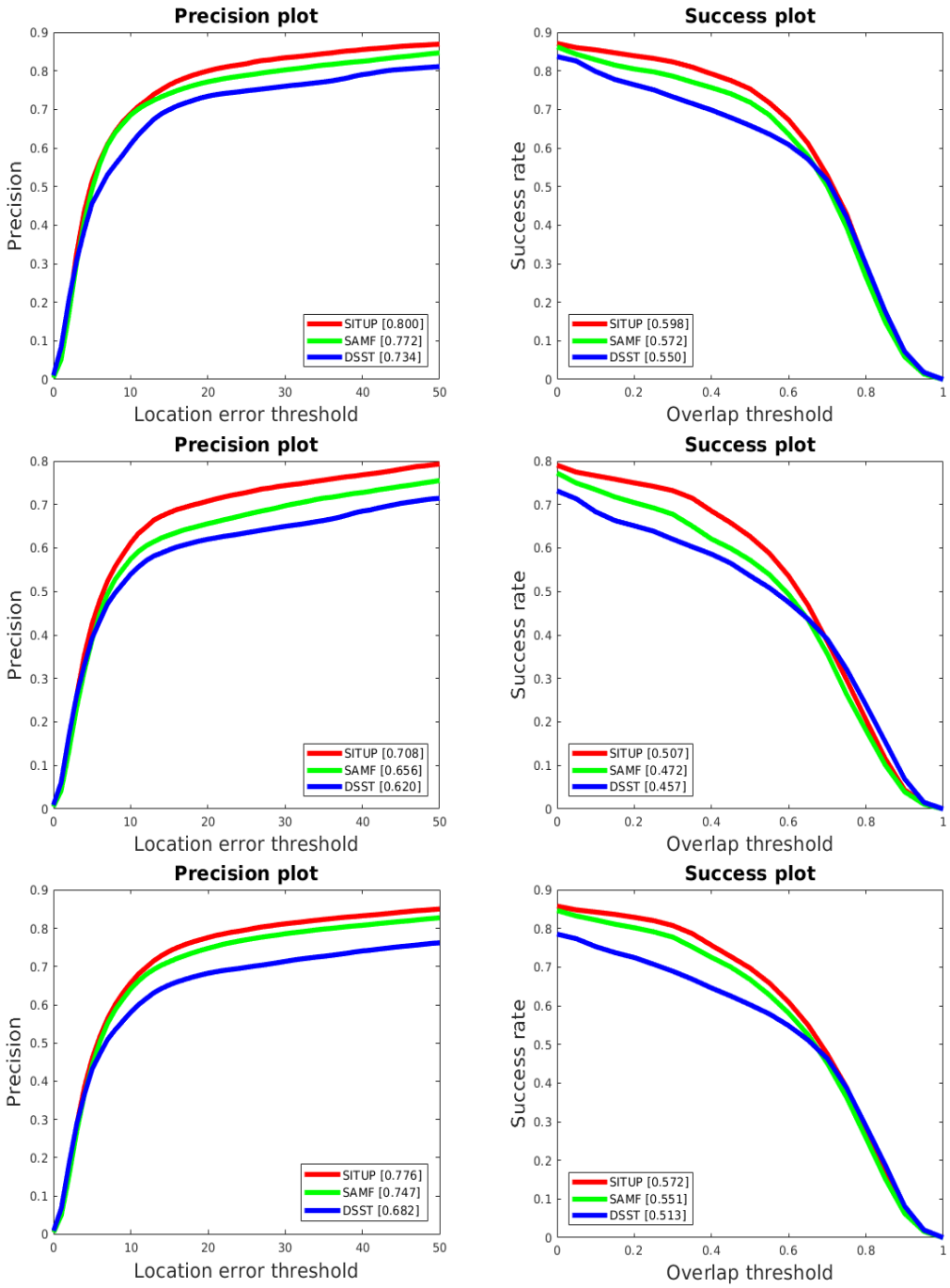

Figure 2.6: The precision plots and success plots of our tracker SITUP and two representative scale adaptive variants of the standard DCF based trackers. From top to bottom, the plots correspond to OTB2013, OTB50, OTB2015, respectively. The left column corresponds to the precision plots and the right column corresponds to the success plots.

Attribute-based comparison

Since SITUP is specifically designed for scale estimation, we compare SITUP with two representative scale adaptive DCF variants on the scale variation attribute of OTB2013, OTB50 and OTB2015. OTB2013, OTB50 and OTB2015 contain 28, 38 and 66 scale variation sequences, respectively, and can be used to evaluate the scale adaptability of our tracker. Table 2.2 summarizes the performance of the trackers in terms of the AUC score and the precision score at a threshold of 20 pixels on the scale variation attribute, and the success plots and precision plots are given in Figure 2.7.

As shown in Table 2.2 and Figure 2.7, our tracker achieves the best performance. Comparing to DSST, which obtains the second best performance on OTB2013, our approach improves the precision score by 3.1%, 11.5% and 10.2% and the AUC score by 1.6%, 7.4% and 6.0% on OTB2013, OTB50 and OTB2015, respectively. The remarkable performance of our tracker shows that our method is superior in scale estimation comparing to other trackers despite the simplicity of our framework.

Table 2.2: Attributed-based comparisons with two scale adaptive variants of standard DCF based trackers on the scale variation (SV) attribute of OTB2013, OTB50, and OTB2015 benchmark sequences. The first and the second best values are highlighted in bold and underlined, respectively.

		OTB2013		OTB50		OTB2015	
	Method	**Prec.**	**AUC**	**Prec.**	**AUC**	**Prec.**	**AUC**
SV	DSST	<u>0.736</u>	<u>0.544</u>	0.580	0.421	0.654	0.482
	SAMF	0.723	0.507	<u>0.637</u>	<u>0.449</u>	<u>0.719</u>	<u>0.507</u>
	SITUP	**0.767**	**0.560**	**0.695**	**0.495**	**0.756**	**0.542**

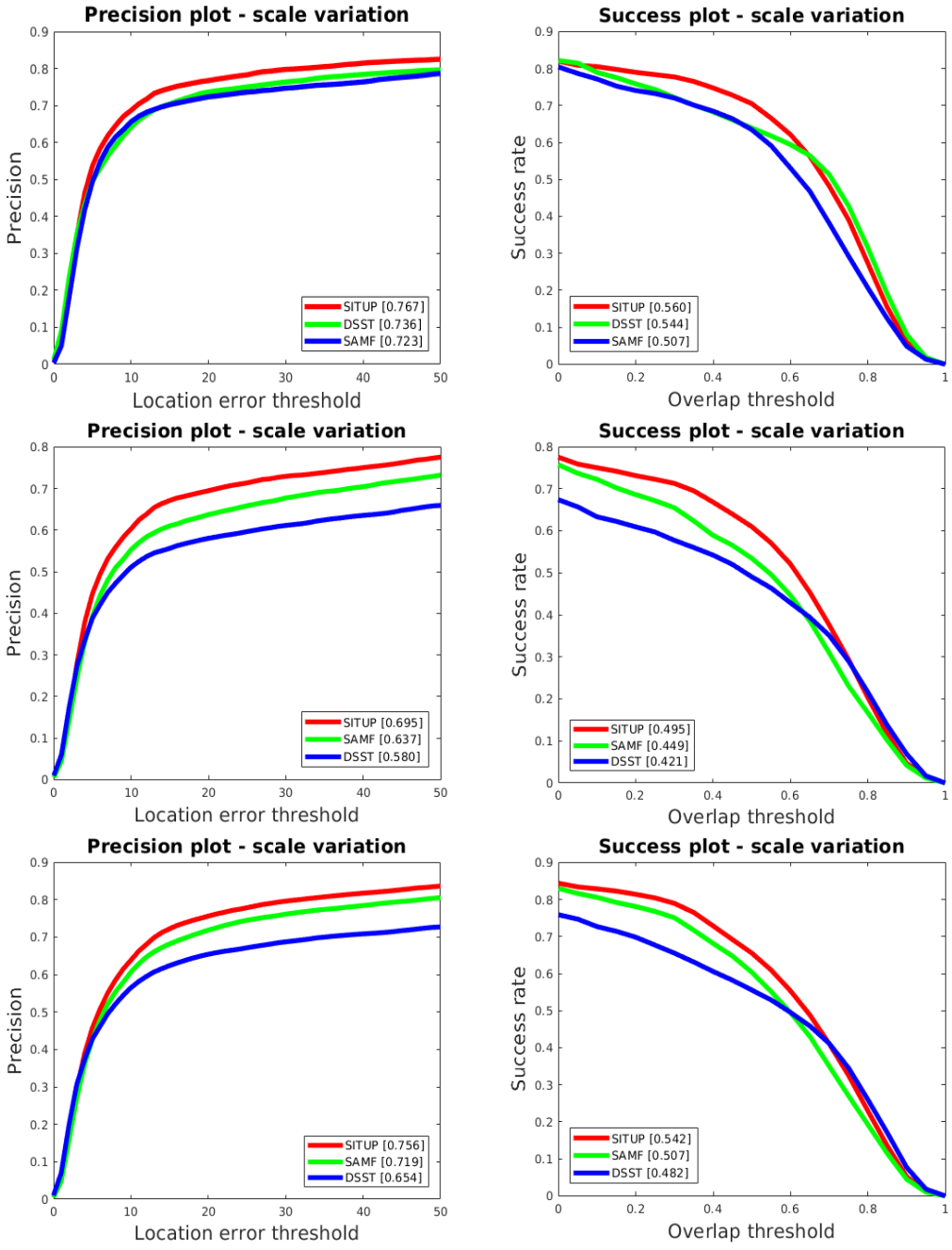

Figure 2.7: The precision plots and success plots of our tracker SITUP and two representative scale adaptive variants of standard DCF based trackers on the scale variation (SV) attribute of OTB2013, OTB50, and OTB2015 benchmark sequences. From top to bottom, the plots correspond to OTB2013, OTB50, OTB2015, respectively. The left column corresponds to the precision plots and the right column corresponds to the success plots.

Comparing to SAMF, which also employs the multi-resolution translation filter framework, our tracker improves the precision score by 4.3%, 5.8% and 3.7% and the AUC score by 5.3%, 4.6% and 3.5% on OTB2013, OTB50 and OTB2015, respectively. The comparison between our tracker and SAMF shows the effectiveness and robustness of the employed criterion APCE in the scale variation scenario.

2.5 Discussion

In this chapter, SITUP, a scale estimation approach, is proposed to address the problem of fixed template size in standard discriminative correlation filter (DCF) based trackers by incorporating a novel criterion named average peak-to-correlation energy (APCE) into the multi-resolution translation filter framework. The concise integration of the APCE measure in an exhaustive scale searching framework makes our scale estimation method generic and easy to be incorporated into any DCF based tracker. Extensive experiments are conducted on the OTB datasets. The comparison between SITUP and other scale adaptive variants of standard DCF based trackers clearly demonstrates the effectiveness of our scale searching strategy and the robustness of the employed criterion APCE. However, the exhaustive scale searching scheme employed in SITUP is computationally demanding when facing large scale variations. Therefore, based on SITUP, we propose a computationally efficient scale adaptive tracker that will be detailed in the next chapter.

3

FAST: Fast and Accurate Scale Estimation for Tracking

The objective of this chapter is to introduce a scale adaptive DCF based tracker that can be used for scale estimation when the computational resource is limited. The multi-scale searching scheme employed in SITUP is computationally demanding when facing large scale variations. Based on SITUP, we investigate different strategies to reduce the computational cost. The resulting system is

named FAST: fast and accurate scale estimation for tracking. Extensive experiments are performed on the tracking benchmark datasets to verify the effectiveness of FAST. In comparison with SITUP, FAST obtains comparable performance while operating at a three times higher frame rate, which validates the effectiveness of the employed computational cost reduction strategy.

As to the organization of this chapter, Section 3.1 introduces the background and motivation of the proposed tracker FAST. Section 3.2 describes our proposed tracker FAST in detail. In particular, the formulation of our baseline tracker multi-channel DCF is treated in Section 3.2.1. Our scale searching scheme is described in Section 3.2.2. The strategies investigated to reduce the computational cost are introduced in Section 3.2.3. The experimental setup and comparison results are presented in Section 3.3. Finally, Section 3.4 concludes the chapter.

3.1 Background

As a basic unit of many time-critical computing systems, a given visual tracker should meet strict time and computational requirements. These requirements are particularly acute in embedded and mobile systems where resources are limited. An ideal tracking approach should be robust regarding scale variations while operating beyond real-time. In Chapter 2, via exploiting the robustness of a novel criterion called the average peak-to-correlation energy, our proposed tracker SITUP meets the demands of scale estimation effectively while providing superior performance over other scale adaptive variants of standard discriminative correlation filter based trackers. However, SITUP is computationally expensive when facing large scale variations.

To address the problem of the computationally expensive exhaustive scale searching scheme, based on SITUP, we investigate strategies to reduce the computational cost. Also, the simple linear interpolation scheme for model updating employed in SITUP is proved to be sub-optimal[23], since only the current frame is taken into account while updating the model and the contribution of

previous frames in model fades out exponentially. Here, we employ a more robust updating scheme presented in reference 23, which considers current frame and all previous frames simultaneously. In comparison with SITUP, FAST obtains comparable performance while operating at a three times higher frame rate, which validates the effectiveness of the employed computational cost reduction strategy.

3.2 Proposed Method

3.2.1 Multi-channel Discriminative Correlation Filters

FAST leverages the multi-channel DCF. The problem of learning an optimal correlation filter can be formulated as training a classifier using all cyclic shifts of a base sample. The feature channel $l \in \{1, 2, ..., c\}$ of the target sample $\mathbf{x}$ is denoted as $\mathbf{x}^l$. The objective of learning the correlation filter $\mathbf{w}$ consisting of one filter $\mathbf{w}^l$ per feature channel can be achieved by minimizing the objective function

$$\varepsilon = \left\| \mathbf{y} - \sum_{l=1}^{c} \mathbf{w}^l \star \mathbf{x}^l \right\|^2 + \lambda \sum_{l=1}^{c} \left\| \mathbf{w}^l \right\|^2, \tag{3.1}$$

where $\star$ denotes the circular correlation and λ is a regularization parameter. The desired output $\mathbf{y}$ is a Gaussian function parameterized by the standard deviation. The objective function Equation (3.1) is minimized by

$$\mathbf{W}^l = \frac{\mathbf{A}^l}{\mathbf{B} + \lambda} = \frac{\overline{\mathbf{Y}}\mathbf{X}^l}{\sum_{l=1}^{c} \overline{\mathbf{X}^l}\mathbf{X}^l + \lambda}, \; l \in \{1, 2, ..., c\}, \tag{3.2}$$

where $\mathbf{W}^l$, $\mathbf{Y}$ and $\mathbf{X}^l$ denote the discrete Fourier transform (DFT) of $\mathbf{w}^l$, $\mathbf{y}$ and $\mathbf{x}^l$, respectively, and $\overline{\mathbf{Y}}$ and $\overline{\mathbf{X}^l}$ denote the complex conjugate of $\mathbf{Y}$ and $\mathbf{X}^l$, respectively.

To adapt to the target appearance changes, the numerator $\mathbf{A}^l$ and the denominator $\mathbf{B}$ of Equa-

tion (3.2) are updated as

$$\mathbf{A}_t^l = (1-\eta)\mathbf{A}_{t-1}^l + \eta\overline{\mathbf{Y}}\mathbf{X}_t^l, \tag{3.3}$$

$$\mathbf{B}_t = (1-\eta)\mathbf{B}_{t-1} + \eta\sum_{l=1}^{c}\overline{\mathbf{X}_t^l}\mathbf{X}_t^l, \tag{3.4}$$

where t is the frame number and η denotes the learning rate. The response map $\mathbf{f}_t$ corresponds to the test sample $\mathbf{z}_t$ can be computed as

$$\mathbf{f}_t = \mathcal{F}^{-1}\left(\frac{\sum_{l=1}^{c}\overline{\mathbf{A}_{t-1}^l}\mathbf{Z}_t^l}{\mathbf{B}_{t-1}+\lambda}\right), \tag{3.5}$$

where $\mathbf{Z}_t^l$ is the DFT of $\mathbf{z}_t^l$. The translation estimation is obtained by localizing the maximum response value within $\mathbf{f}_t$.

3.2.2 The Proposed Scale Searching Scheme of FAST

The fluctuation and the peak value of the response map can reveal the confidence level of the tracking result[34,36]. The ideal response map should have only one sharp peak at the true target location while being close to zero in all other areas, which will result in higher localization accuracy and associated APCE measure. On the contrary, a fierce fluctuation of the response map will result in a significant decrease of the APCE measure when encountering challenging scenarios, such as severe occlusion and illumination variation. Hence, the APCE measure, which takes both the peak value and the fluctuation of the response map into consideration, will be more robust than the naïve maximum response value when facing such challenges.

At the testing stage of frame t, k patches with their sizes in $\{\mathbf{h}_{t-1}b|b \in \{s_1, s_2, ..., s_k\}\}$ are extracted and resized to a fixed template size through bilinear interpolation, where $\mathbf{h}_{t-1}$ denotes the size of the searching window of the previous frame. The response map corresponds to the resized

patch is denoted as $\mathbf{f}_{t,b}$, and the associated APCE measure can be computed as

$$\mathrm{APCE}_{t,b} = \frac{|\mathbf{f}_{t,b}^{\max} - \mathbf{f}_{t,b}^{\min}|^2}{\mathrm{mean}\left(\sum_{m,n}(\mathbf{f}_{t,b}^{m,n} - \mathbf{f}_{t,b}^{\min})^2\right)}, \tag{3.6}$$

where $\mathbf{f}_{t,b}^{\max}$, $\mathbf{f}_{t,b}^{\min}$, and $\mathbf{f}_{t,b}^{m,n}$ denote the maximum, the minimum and the m-th row n-th column elements of the response map $\mathbf{f}_{t,b}$, respectively. The scale estimation is obtained by searching for the response map with the highest APCE measure as

$$\mathit{scale} = \arg\max_{b} \mathrm{APCE}_{t,b}. \tag{3.7}$$

The translation estimation is obtained by localizing the maximum response value within the response map with the highest APCE measure.

3.2.3 Computational Cost Reduction

As shown in Section 3.2.1, the computational cost is dominated by the DFT, and the number of the DFT computations scales linearly with the feature dimension c. We propose to carry out feature dimension reduction to reduce the required number of DFT computations. Our feature dimension reduction strategy is based on standard principal component analysis (PCA).

By the linearity of the Fourier transform, the numerator of Equation (3.2), given by Equation (3.3), can be rewritten as

$$\mathbf{A}_t^l = \overline{\mathbf{Y}}\mathcal{F}(\mathbf{g}_t^l), \tag{3.8}$$

where the target template is denoted as

$$\mathbf{g}_t = (1-\eta)\mathbf{g}_{t-1} + \eta\mathbf{x}_{t,\mathit{scale}}. \tag{3.9}$$

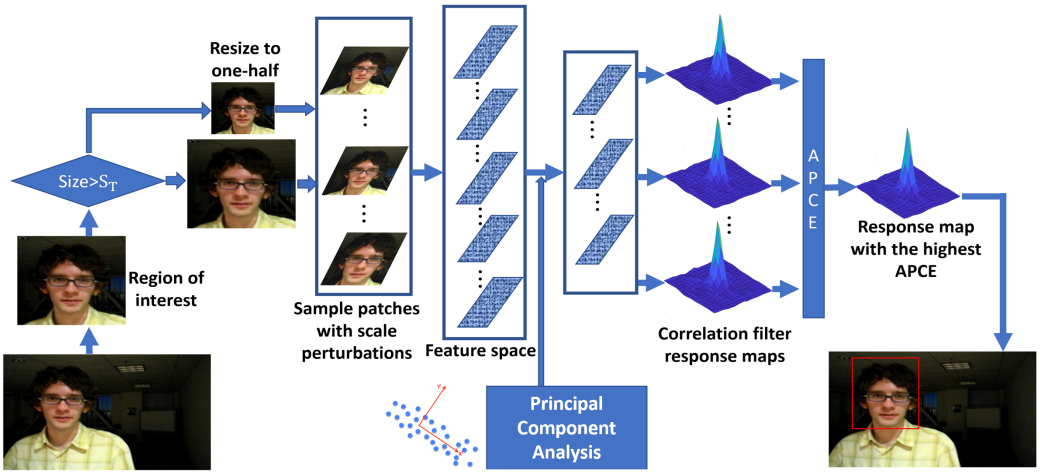

Figure 3.1: The tracking pipeline of FAST. First, the target sample is resized to one-half of the original size if the target sample size in the initial frame is higher than a defined threshold. Second, patches with different scales are sampled and the corresponding features are computed, and then the standard principal component analysis is employed to reduce the feature dimension to yield computational efficiency. Last, the response maps are computed and the response map with the highest APCE measure is chosen as the one with the best scale estimation. The translation estimation is obtained by localizing the maximum response value within the response map with the highest APCE measure.

The target template is employed to construct the projection matrix $\mathbf{P} \in \tilde{c} \times c$, which projects the feature onto a $\tilde{c}$-dimensional subspace while preserving the useful information. Under the orthogonality constraint $\mathbf{P}_t\mathbf{P}_t^{\mathrm{T}} = \mathbf{I}$, the projection matrix can be obtained by minimizing the reconstruction error of the target template as

$$\varepsilon = \left\| \mathbf{g}_t - \mathbf{P}_t^{\mathrm{T}}\mathbf{P}_t\mathbf{g}_t \right\|^2. \tag{3.10}$$

A solution to Equation (3.10) can be achieved by performing an eigenvalue decomposition of

$$\mathbf{O}_t = \mathbf{g}_t\mathbf{g}_t^{\mathrm{T}}, \tag{3.11}$$

which denotes the auto-correlation matrix of the target template. The rows of $\mathbf{P}_t$ are set as the $\tilde{c}$

eigenvectors, which correspond to the largest eigenvalues. The correlation filter is updated as

$$\tilde{\mathbf{A}}_t^l = \overline{\mathbf{Y}}\tilde{\mathbf{G}}_t^l, \tag{3.12}$$

$$\tilde{\mathbf{B}}_t = (1-\eta)\tilde{\mathbf{B}}_{t-1} + \eta \sum_{l=1}^{\tilde{c}} \overline{\tilde{\mathbf{X}}_{t,scale}^l}\tilde{\mathbf{X}}_{t,scale}^l, \tag{3.13}$$

where the compressed training sample can be written as

$$\tilde{\mathbf{X}}_{t,scale} = \mathcal{F}\{\mathbf{P}_t \mathbf{x}_{t,scale}\}, \tag{3.14}$$

and the compressed target template is denoted as

$$\tilde{\mathbf{G}}_t = \mathcal{F}\{\mathbf{P}_t \mathbf{g}_t\}. \tag{3.15}$$

The compressed test sample patch is computed as

$$\tilde{\mathbf{Z}}_{t,b} = \mathcal{F}\{\mathbf{P}_{t-1} \mathbf{z}_{t,b}\}. \tag{3.16}$$

Equation (3.5) can then be rewritten as

$$\mathbf{f}_{t,b} = \mathcal{F}^{-1}\left(\frac{\sum_{l=1}^{\tilde{c}} \overline{\tilde{\mathbf{A}}_{t-1}^l}\tilde{\mathbf{Z}}_{t,b}^l}{\tilde{\mathbf{B}}_{t-1} + \lambda}\right), \tag{3.17}$$

where $\tilde{\mathbf{Z}}_{t,b}^l$ denotes the feature channel l of the compressed test sample patch $\tilde{\mathbf{Z}}_{t,b}$.

Benefiting from the robustness of the APCE criterion, the significant performance gain provides the flexibility to reduce the size of the performed DFTs. Here, the target sample is resized to one-half of the original size if the target sample size in the initial frame is higher than a defined threshold.

3.3 Experiments

3.3.1 Implementation Details

The regularization parameter is set as $\lambda = 0.08$, and the learning rate is set as $\eta = 0.004$. The standard deviation of the desired correlation output is set to be 1/10 of the target size. The padding ratio is set as 1.5 to make the search window 2.5 times the target size in terms of both width and height, which is typical in the existing literature.

For the scaling pool, we employ the same setting $\mathbf{S} = \{0.985, 0.99, 0.995, 1.0, 1.005, 1.01, 1.015\}$ as in SITUP. The features we employ are computed by augmenting HoG, and grayscale pixel values. In terms of HOG, a 31-dimensional variant of HoG is employed as in reference 19. The grayscale features are normalized to the range $[-\frac{1}{2}, \frac{1}{2}]$. The augmented 32-dimensional feature will be projected onto an 18-dimensional subspace as in reference 37. Each extracted feature channel of the sample patch is weighted by a cosine window to address the boundary effect.

All tracking algorithms are implemented with Matlab on an Intel Core i7-7700K 4.5 GHz CPU with 16 GB RAM. A GeForce GTX 1080 GPU and MatConvNet 1.0-beta25 [35] are used to reproduce the results of the trackers that require GPU implementation. Parameters of our tracking algorithm are fixed in all experiments. The parameters of other trackers are set according to the values in their original code.

3.3.2 Datasets and Metrics

As in SITUP, all the experiments are performed on the online tracking benchmark (OTB) datasets (OTB2013, OTB50 and OTB2015), and the same evaluation metrics are employed. All the numbers and plots are obtained using the standard OTB toolkit. The detailed description of the benchmark datasets and evaluation metrics can be found in Section 2.4.2. The running speed of each

tracking algorithm is obtained by executing the tracker on the OTB2013 dataset, and dividing the total frames of OTB2013 by the total running time in seconds.

3.3.3 Performance Comparison

Overall comparison

To validate the overall performance of FAST, two trackers (SITUP and FAST) are executed on the OTB datasets. The overall performance of the trackers is summarized in Table 3.1 in terms of the AUC score and the precision score for a threshold of 20 pixels. The success plots and precision plots are presented in Figure 3.2.

As shown in Table 3.1 and Figure 3.2, on OTB2013, FAST outperforms SITUP by 2.4% and 2.3% on precision score and AUC score respectively, and underperforms SITUP by 3.2% and 1.6% on precision score and AUC score by 0.7% and 0.1% on more challenging OTB50 and OTB2015, respectively. Comparing to SITUP, FAST obtains comparable performance while operating at a three times higher frame rate, which validates the effectiveness of the employed computational cost reduction strategy.

Table 3.1: Comparisons between SITUP and FAST on OTB2013, OTB50 and OTB2015 benchmark sequences in terms of the area-under-curve (AUC) score and the precision score (Prec.) at a standard threshold of 20 pixels used in the literature. The first and the second best values are highlighted in bold and underlined, respectively.

		OTB2013		OTB50		OTB2015	
Method	**Speed**	**Prec.**	**AUC**	**Prec.**	**AUC**	**Prec.**	**AUC**
SITUP	<u>32.35</u>	<u>0.800</u>	<u>0.598</u>	**0.708**	**0.507**	**0.776**	**0.572**
FAST	**91.73**	**0.824**	**0.621**	<u>0.676</u>	<u>0.500</u>	<u>0.760</u>	<u>0.571</u>

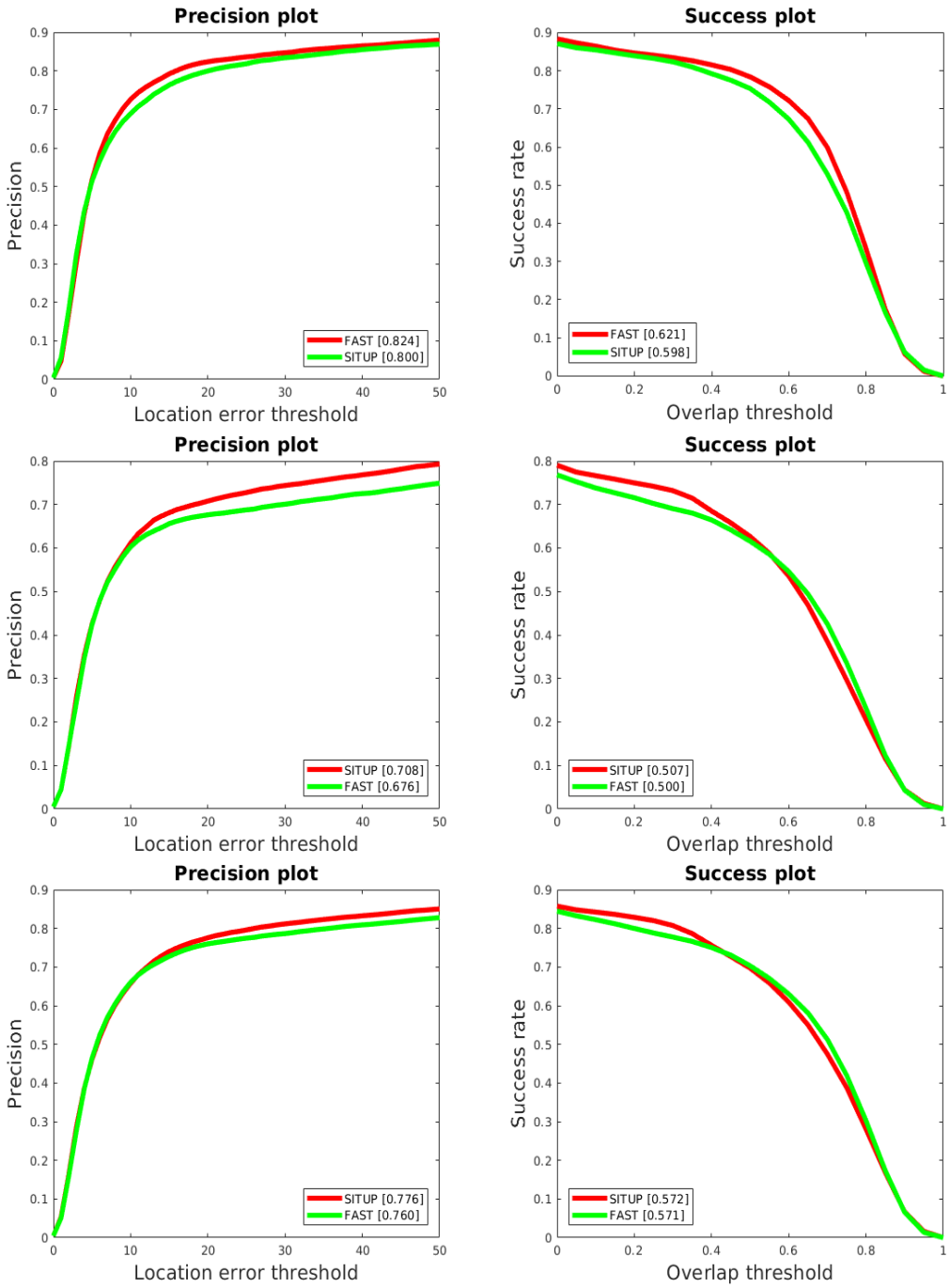

Figure 3.2: The precision plots and success plots for the overall performance of SITUP and FAST on OTB datasets. From top to bottom, the plots correspond to OTB2013, OTB50 and OTB2015, respectively. The left column corresponds to the precision plots, and the right column corresponds to the success plots.

Attribute-based comparison

Since FAST is designed to address the scale variation challenges, to validate the scale adaptability of our proposed tracker FAST, we compare FAST with SITUP on the scale variation attribute of OTB datasets. The performance of the trackers on the scale variation attribute is summarized in Table 3.2 in terms of the AUC score and the precision score for a threshold of 20 pixels. The success plots and precision plots are presented in Figure 3.3.

As shown in Table 3.2 and Figure 3.3, in comparison with SITUP, FAST obtains the best AUC score and precision score on OTB2013 and comparable AUC score on OTB50 and OTB2015, which evidences the effectiveness of the new model updating scheme. On the more challenging OTB50 and OTB2015 datasets, FAST obtains the second best precision scores that are 4.7% and 3.5% lower than the results obtained by SITUP, respectively. It is likely that the performance gap mainly comes from the computational cost reduction strategy since we resize the target sample to one-half when the input target sample size in the initial frame is higher than a defined threshold.

Table 3.2: Comparisons between SITUP and FAST on the scale variation (SV) attribute of OTB2013, OTB50 and OTB2015 benchmark sequences in terms of the area-under-curve (AUC) score and the precision score (Prec.) at a standard threshold of 20 pixels used in the literature. The first and the second best values are highlighted in bold and underlined, respectively.

		OTB2013		**OTB50**		**OTB2015**	
	Method	**Prec.**	**AUC**	**Prec.**	**AUC**	**Prec.**	**AUC**
SV	SITUP	<u>0.767</u>	<u>0.560</u>	**0.695**	**0.495**	**0.756**	**0.542**
	FAST	**0.772**	**0.579**	<u>0.648</u>	<u>0.478</u>	<u>0.721</u>	<u>0.533</u>

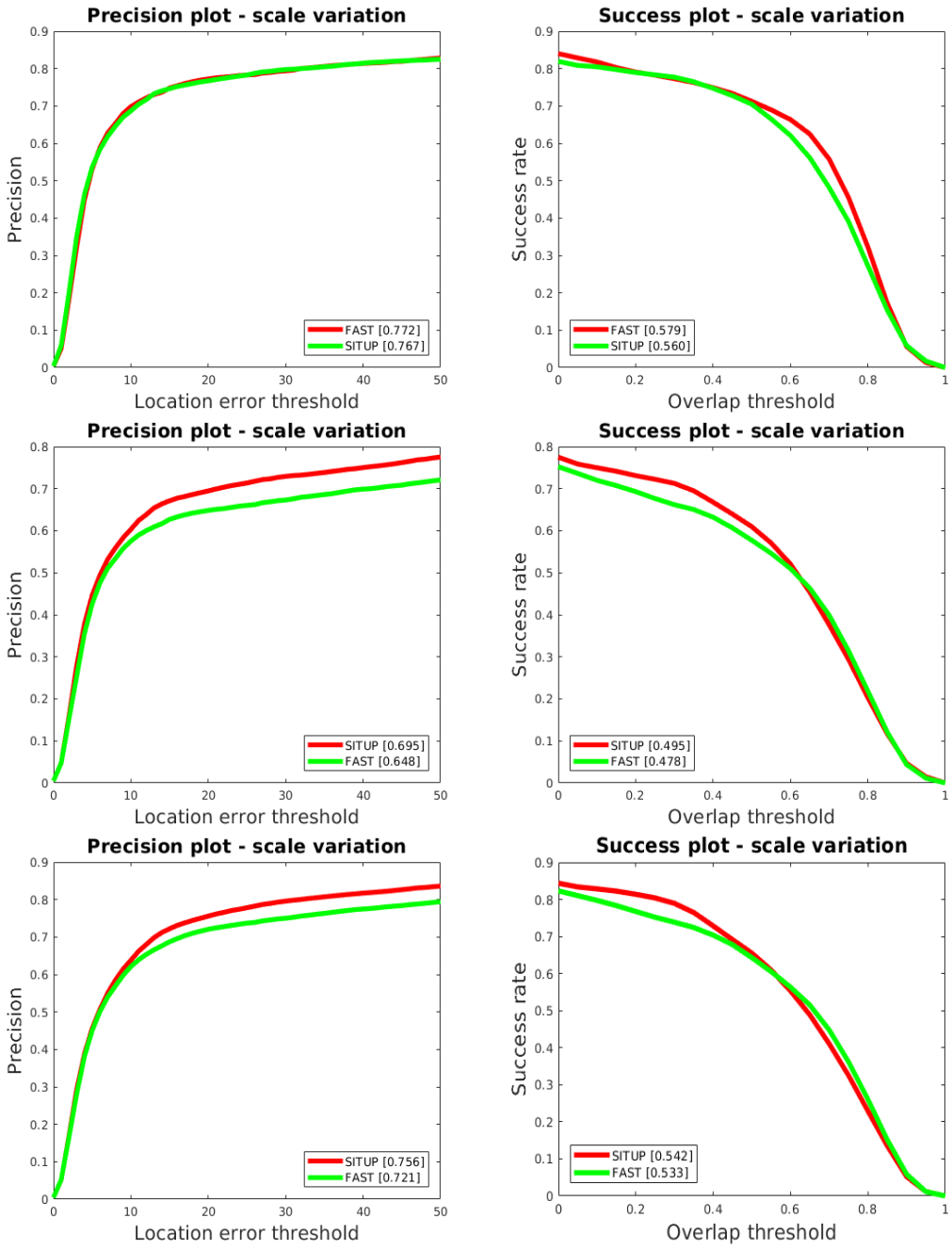

Figure 3.3: The precision plots and success plots of SITUP and FAST on the scale variation (SV) attribute of OTB2013, OTB50, and OTB2015 benchmark sequences. From top to bottom, the plots correspond to OTB2013, OTB50, OTB2015, respectively. The left column corresponds to the precision plots, and the right column corresponds to the success plots.

3.3.4 Internal Analysis of the Proposed Approach

To investigate the impact of our proposed scale estimation module and computational cost reduction module, we propose three different variants of our proposed tracker FAST: the baseline tracker (DCF), DCF with only the scale estimation module denoted by DCF_se, and DCF with only the computational cost reduction module denoted by DCF_ccr. The performance of the proposed FAST tracker and its three variants on OTB2015 are summarized in Table 3.3.

Table 3.3: Overall comparisons of FAST tracker and its three variants on OTB2015 datasets. The first and the second best values are highlighted in bold and underlined, respectively.

Dataset	Metric	DCF	DCF_se	DCF_ccr	FAST
OTB2015	AUC	0.486	<u>0.558</u>	0.485	**0.571**
	Prec.	0.692	<u>0.729</u>	0.696	**0.760**
	Speed	<u>160.5</u>	48.86	**266.13**	90.08

As shown in the comparison between DCF_se and DCF along with the comparison between FAST and DCF_ccr, our proposed scale estimation module improves both the AUC score and the precision score significantly, which evidences the effectiveness of our proposed scale estimation module. The comparison between DCF_ccr and DCF together with the comparison between FAST and DCF_se show that our proposed computational cost reduction module can improve the tracking speed remarkably without sacrificing the accuracy. Hence, both the scale estimation module and the computational cost reduction module are key parts of the proposed FAST tracker.

3.4 Discussion

In this chapter, FAST, a computationally efficient scale adaptive tracker is proposed to address the problem of expensive computation cost in SITUP by investigating different computation cost reduction strategies. In comparison with SITUP, FAST obtains a threefold gain in speed while maintaining a comparable tracking performance. Benefiting from its computational efficiency, FAST

can be employed in (*e.g.,* tracking malicious objects using unmanned aerial vehicles). Despite the promising tracking performance of SITUP and FAST, the employed multi-scale searching scheme is not born with aspect ratio adaptability and will result in inferior tracking performance facing aspect ratio variation challenges. To address the aspect ratio variation challenges, we design a scale searching scheme to enable aspect ratio adaptability, which will be detailed in the next chapter.

4

TARA: Tracking with Aspect Ratio Adaptability

The objective of this chapter is to introduce a scale searching scheme with aspect ratio adaptability to address the aspect ratio variation challenges in tracking. The multi-scale searching framework employed in SITUP and FAST is not born with aspect ratio adaptability, and thus will result in tracking failures if the aspect ratio of the target object varies significantly. To address the scale es-

timation problem and enable aspect ratio adaptability, we propose to employ a group of discriminative correlation filters to localize the boundaries of the target object. Inspired by the success of convolutional neural networks (CNNs) features on object detection [38–41], we propose to exploit the rich feature hierarchies in CNNs for robust visual tracking. The resulting system is named TARA: tracking with aspect ratio adaptability. Experiments on benchmark datasets demonstrate TARA can accommodate scale and aspect ratio variation challenges effectively.

As to the organization of this chapter, Section 4.1 introduces the background and the motivation of our proposed tracker TARA. Section 4.2 details how TARA addresses the scale variation challenges and enables the aspect ratio adaptability. In particular, the formulation of the baseline tracker is described in Section 4.2.1. The proposed scale searching scheme is described in Section 4.2.2, and the details about the employed features are described in Section 4.2.3. The experimental setup and comparison results are presented in Section 4.3. Section 4.4 concludes this chapter.

4.1 Background

To incorporate scale estimation into a standard DCF based tracker, SITUP and FAST have been proposed, in which a multi-scale searching scheme is employed. However, the employed multi-scale searching scheme is not born with aspect ratio adaptability, since only one scale factor is employed to denote the change of the target size, and the aspect ratio is fixed as in the initial frame. The aspect ratio variation challenges frequently appear due to interfering challenging factors, such as in-plane/out-of-plane rotation and deformation, and has a severe influence on the tracking performance. As shown in Figure 4.1, the aspect ratio of the target object changes significantly as the transformer transfers to a car. SITUP cannot accommodate the significant aspect ratio variation of the transformer and, hence, yields inferior tracking performance.

Figure 4.1: A failure case of SITUP facing the aspect ratio variation challenge. Example frames shown are from *Trans* sequence that is selected from the online tracking benchmark datasets to involve the aspect ratio variation challenge.

Despite the advancement in scale adaptive DCF based trackers [24,25,27,36,42] with fixed aspect ratio, few studies have been proposed to address the aspect ratio variation problem. To address the aspect ratio variation challenge, Huang *et al.* propose a kernelized correlation filter with detection proposals (KCFDP) [43] to enable aspect ratio adaptability by employing a proposal generator called EdgeBox [44]. However, the detection proposal method EdgeBox employed in KCFDP tends to miss targets in the presence of various factors, such as background clutter, since the EdgeBox generates proposals that favor bounding boxes with many internal contour segments, and thus results in inferior performance.

Recently, inspired by the success of CNN on object detection and recognition, several trackers based on the CNN features are proposed in references 45 and 46. Qi *et al.* propose a method named hedged deep tracking (HDT) [45] to incorporate CNN features into discriminative correlation filter to improve the robustness and accuracy. Further, Ma *et al.* equip HDT with a scale estimation module and the resulting system is named CF2 [46]. Empirical studies show that the performance of CNN based trackers surpass that of hand-crafted features (*e.g.*, HOG [17], color names [47], and scale-invariant feature transform [48]) in precision score. However, these methods cannot adapt to aspect ratio variations either and also suffer from inferior performance in the face of aspect ratio variation challenges.

The main contributions of this chapter are three-fold. First, a scale searching scheme is proposed

to enable aspect ratio adaptability and to obtain robust and accurate scale estimation by localizing the boundaries of the target object. Second, rich deep hierarchical convolutional features of deep CNNs that encode the target object appearance information with different levels of abstraction are exploited to improve the tracking accuracy and robustness when facing different scenarios. Third, comprehensive evaluation using benchmark datasets validates the effectiveness of the proposed tracker TARA.

4.2 Proposed Method

4.2.1 Discriminative Correlation Filter

As in FAST, TARA leverages the multi-channel DCF. Given the feature channel $k \in \{1, 2, ..., g\}$ of the target sample $\mathbf{x}$ represented as $\mathbf{x}^k$, the DCF $\mathbf{w}$ consisting of one filter $\mathbf{w}^k$ per feature channel can be learned using all cyclic shifts of a base sample via minimizing the objective function

$$\varepsilon = \left\| \mathbf{y} - \sum_{k=1}^{g} \mathbf{w}^k \star \mathbf{x}^k \right\|^2 + \lambda \sum_{k=1}^{g} \left\| \mathbf{w}^k \right\|^2, \tag{4.1}$$

where $\star$ denotes the circular correlation and λ is a regularization parameter that controls overfitting. The desired output $\mathbf{y}$ is a Gaussian function parameterized by the standard deviation. The closed-form solution to Equation (4.1) can be written as

$$\mathbf{W}^k = \frac{\mathbf{A}^k}{\mathbf{B} + \lambda} = \frac{\overline{\mathbf{Y}}\mathbf{X}^k}{\sum_{k=1}^{g} \overline{\mathbf{X}^k}\mathbf{X}^k + \lambda}, \; k \in \{1, 2, ..., g\}, \tag{4.2}$$

where $\mathbf{X}^k$, $\mathbf{Y}$ and $\mathbf{W}^k$ denote the DFT of $\mathbf{x}^k$, $\mathbf{y}$ and $\mathbf{w}^k$, respectively, and $\overline{\mathbf{X}^k}$ and $\overline{\mathbf{Y}}$ denote the complex conjugate of $\mathbf{X}^k$ and $\mathbf{Y}$, respectively. To adapt to the target appearance changes, $\mathbf{A}^k$ and $\mathbf{B}$

in Equation (4.2) are updated as

$$\mathbf{A}_t^k = (1-\eta)\mathbf{A}_{t-1}^k + \eta\overline{\mathbf{Y}}\mathbf{X}_t^k, \tag{4.3}$$

$$\mathbf{B}_t = (1-\eta)\mathbf{B}_{t-1} + \eta\sum_{k=1}^{g}\overline{\mathbf{X}_t^k}\mathbf{X}_t^k, \tag{4.4}$$

where t denotes the frame number and η denotes the learning rate. The response map $\mathbf{f}_t$ corresponds to the test sample $\mathbf{z}_t$ can be computed as

$$\mathbf{f}_t = \mathcal{F}^{-1}\left(\frac{\sum_{k=1}^{g}\overline{\mathbf{A}_{t-1}^k}\mathbf{Z}_t^k}{\mathbf{B}_{t-1}+\lambda}\right), \tag{4.5}$$

where $\mathbf{Z}_t^k$ is the DFT of $\mathbf{z}_t^k$. The translation estimation is obtained by localizing the maximum value within $\mathbf{f}_t$.

4.2.2 Scale Searching

In a standard 2D DCF, the target object state $\mathbf{S}$ is characterized by the center location (x, y) and size (w, h) of a bounding box and denoted as

$$\mathbf{S} = [x, y, w, h]. \tag{4.6}$$

Here, as shown in Figure 4.2, we characterize the state of a target object with the boundaries of the corresponding axis-aligned bounding box as $[b_l, b_r, b_t, b_b]$, where b_l and b_r correspond to the horizontal coordinates of the left and right boundaries, respectively, and b_t and b_b correspond to the vertical coordinates of the top and bottom boundaries, respectively.

It should be noted that our employed state representation can be easily transformed to the state

representation employed in the standard 2D DCF as

$$
\begin{aligned}
x &= (b_l + b_r)/2, \\
y &= (b_t + b_b)/2, \\
w &= b_r - b_l, \\
h &= b_t - b_b.
\end{aligned}
\tag{4.7}
$$

Figure 4.2: The comparison between the state representation employed in TARA and the standard state representation employed in a standard DCF. The left figure corresponds to the standard state representation employed in a standard 2D DCF, in which the bounding box is denoted using the width, height, and the center coordinates of the bounding box. The right figure denotes our employed state representation, in which the horizontal coordinates of the left and right boundaries, and the vertical coordinates of the top and bottom boundaries are employed to denote the bounding box.

In our method, as shown in Figure 4.3, in addition to a standard 2D DCF, which is employed to obtain an initial translation estimation, four 1D DCFs will be learned and incorporated to refine the boundaries of the bounding box to enable scale and aspect ratio adaptability. For the sake of

simplicity in notation, we assume that the number of the feature channel g is 1. Let us consider the 1D DCF that corresponds to the left boundary as an example, given a bounding box of the target with center (x, y) and size (w, h), a left boundary region $\mathbf{r}_l$ centered at $(x - w/2, y)$ with width w_l and height h_l, which will be cropped. The cropped region $\mathbf{r}_l$ can be represented row-wise as

$$\mathbf{r}_l = [\mathbf{r}_{l,1}, \mathbf{r}_{l,2}, ..., \mathbf{r}_{l,h_l}]^{\mathrm{T}} \tag{4.8}$$

and $\mathbf{r}_{l,h_l}$ will be treated as the h_l-th feature channel of the target sample $\mathbf{r}_l$. The DCF that corresponds to the left boundary is denoted as $\mathbf{w}_l$ and can be learned by minimizing the following objective function

$$\varepsilon_l = \left\| \mathbf{y}_l - \sum_{m=1}^{h_l} \mathbf{w}_{l,m} \star \mathbf{r}_{l,m} \right\|^2 + \lambda \sum_{m=1}^{h_l} \left\| \mathbf{w}_{l,m} \right\|^2, \tag{4.9}$$

where $\mathbf{y}_l$ is a 1D Gaussian shaped label. Similarly, with the difference being top and bottom boundary regions represented column-wise, the other three 1D DCFs can be learned to localize the top, bottom, and right boundaries, respectively.

When a new frame t is acquired, a search region will be cropped first according to the estimated center location (x_{t-1}, y_{t-1}) and size (w_{t-1}, h_{t-1}) from the previous frame, and then convolved with the standard 2D DCF to obtain the initial estimation of the center location $(x_{t,init}, y_{t,init})$ in frame t. The resulting estimated center location in the new frame together with the estimated size from the previous frame will be treated as an initial state estimation of the target in frame t and denoted as

$$\mathbf{S}_{t,init} = [x_{t,init}, y_{t,init}, w_{t-1}, y_{t-1}]. \tag{4.10}$$

Then, to enable scale and aspect ratio adaptability, according to the initial state estimation, four boundary regions will be cropped and convolved with corresponding DCFs to obtain the state representation as the locations of the corresponding boundaries, which is then transformed and

represented as

$$\mathbf{S}_{t,new} = [x_{t,new}, y_{t,new}, w_{t,new}, y_{t,new}]. \tag{4.11}$$

To improve the accuracy and robustness, we will compare the new tracking result with the initial tracking result by convolving the regions characterized by $\mathbf{S}_{t,init}$ and $\mathbf{S}_{t,new}$ with the standard 2D DCF, respectively. The result with the highest associated confidence score will be denoted as $\mathbf{S}_{t,good}$, and treated as the state estimation for the current frame.

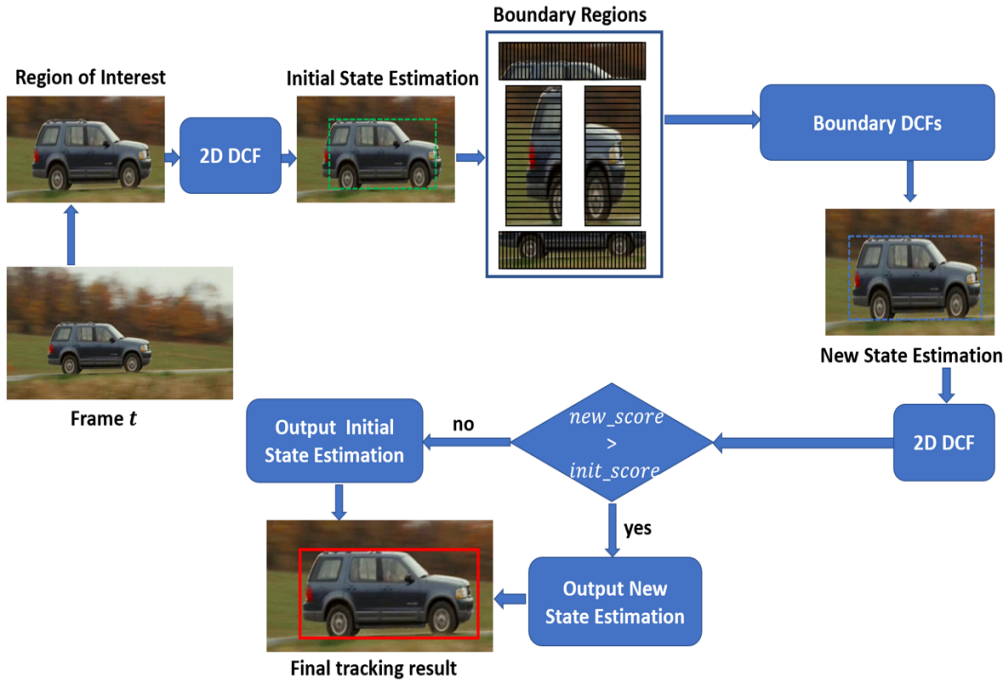

Figure 4.3: The tracking pipeline of TARA. The 2D discriminative correlation filter is employed to obtain the initial state estimation. Based on the initial state estimation, the boundary regions are cropped and four 1D discriminative correlation filters are used to obtain a candidate state estimation with aspect ratio adaptability. The candidate state estimation is further verified via the 2D discriminative correlation filter, and the one between initial state estimation and candidate state estimation with a higher response value will be selected as the final state estimation for the current frame.

4.2.3 Hierarchical Convolutional Features

Inspired by references 45 and 46, to improve the accuracy and robustness of our tracker, we propose to exploit the rich hierarchical features of CNNs. CNNs trained on large object recognition datasets consist of multiple convolutional layers, which encode target appearance information with different levels of abstraction. Features from deeper convolutional layers encode richer category-level semantic information and are more robust to large appearance variations. However, spatial resolutions obtained within deeper feature layers are too coarse to provide accurate localization. In contrast, features from earlier layers capture more spatial details and are more generic than discriminative. Hence, tracking methods based on features from earlier layers can provide accurate localization and are more likely to fail in challenging scenarios.

Here, we learn DCFs using features from both deeper and earlier convolutional layers and compute the corresponding response maps. All the response maps will be resized to a fixed size and represented as $\mathbf{f}^j$ with $j \in \{1, 2, \ldots, n\}$, where n is the number of convolutional layers employed. The final response map $\mathbf{f}$ will be computed adaptively as

$$\mathbf{f} = \sum_{j=1}^{n} \alpha^j \mathbf{f}^j, \tag{4.12}$$

where α^j denotes the weight assigned to the response map corresponding to the j-th convolutional layer.

4.3 Experiments

4.3.1 Implementation Details

For the standard 2D DCF, we employ the outputs of the *conv*3-4, *conv*4-4, and *conv*5-4 convolutional layers of VGG-Net-19 [49] as the convolution features, and the weights for the corresponding

response maps are assigned as 1, 0.5 and 0.25 as in reference 50, respectively. Regarding the four 1D DCFs, we employ the outputs of *conv*4-4 and *conv*5-4 as features, and the weights for the corresponding response maps are assigned as 1 and 1, respectively. The regularization parameter λ, and the learning rate η are set as 0.0001 and 0.01, respectively.

4.3.2 Datasets and Metrics

All trackers are quantitatively evaluated on the online tracking benchmark (OTB) datasets (OTB2013, OTB50 and OTB2015). OTB2013 and OTB2015 contain 50 and 100 video sequences, respectively. Among OTB2015, 50 difficult and representative ones are selected and represented as OTB50. The sequences in OTB datasets are categorized by way of 11 attributes. For example, one sequence is annotated with scale variation (SV) challenge if the ratio between the size of the bounding box of the first frame and any following frame exceeds the range $[1/2, 2]$. As in SITUP, three standard evaluation metrics are employed including precision score, success rate and speed. The detailed description of the benchmark datasets and evaluation metrics can be found in Section 2.4.2.

Existing approaches to SV cannot fully adapt to the situation as they consider only the size change of the bounding box. To further validate the aspect ratio adaptability of our method, we additionally annotate the OTB datasets with aspect ratio variation (ARV) attribute as in reference 43. One frame is marked as undergoing aspect ratio change if the relative ratio between the aspect ratio of the current frame and the aspect ratios of all its 30 previous frames exceeds the range $\left[1/\sqrt{2}, \sqrt{2}\right]$. One sequence will be annotated with ARV challenge if more than 10% of its frames undergo aspect ratio change.

4.3.3 Performance Comparison

Overall Comparison

To compare the overall performance of TARA to other representative DCF based trackers, we execute four trackers (KCFDP, CenTracker, sCenTracker and TARA) on the OTB datasets (OTB2013, OTB50 and OTB2015). CenTracker denotes our employed 2D discriminative correlation filter, sCenTracker denotes the CenTracker equipped with the multi-scale searching scheme for scale estimation. Table 4.1 summarizes the overall performance of the trackers in terms of AUC score and precision score for a threshold of 20 pixels (a typical parameter used in the literature). The success plots and precision plots are presented in Figure 4.4.

Table 4.1: Comparisons between TARA with three discriminative correlation filter based trackers on OTB2013, OTB50, and OTB100 benchmark sequences. The first and the second best values are highlighted in bold and underlined, respectively.

		OTB2013		**OTB50**		**OTB2015**	
Method	**Speed**	**Prec.**	**AUC**	**Prec.**	**AUC**	**Prec.**	**AUC**
KCFDP	**30.27**	0.770	0.566	0.670	0.483	0.728	0.538
CenTracker	<u>9.82</u>	0.858	0.582	0.775	0.496	0.814	0.547
sCenTracker	1.93	<u>0.882</u>	<u>0.593</u>	<u>0.788</u>	<u>0.521</u>	<u>0.831</u>	<u>0.570</u>
TARA	2.62	**0.891**	**0.666**	**0.795**	**0.577**	**0.840**	**0.627**

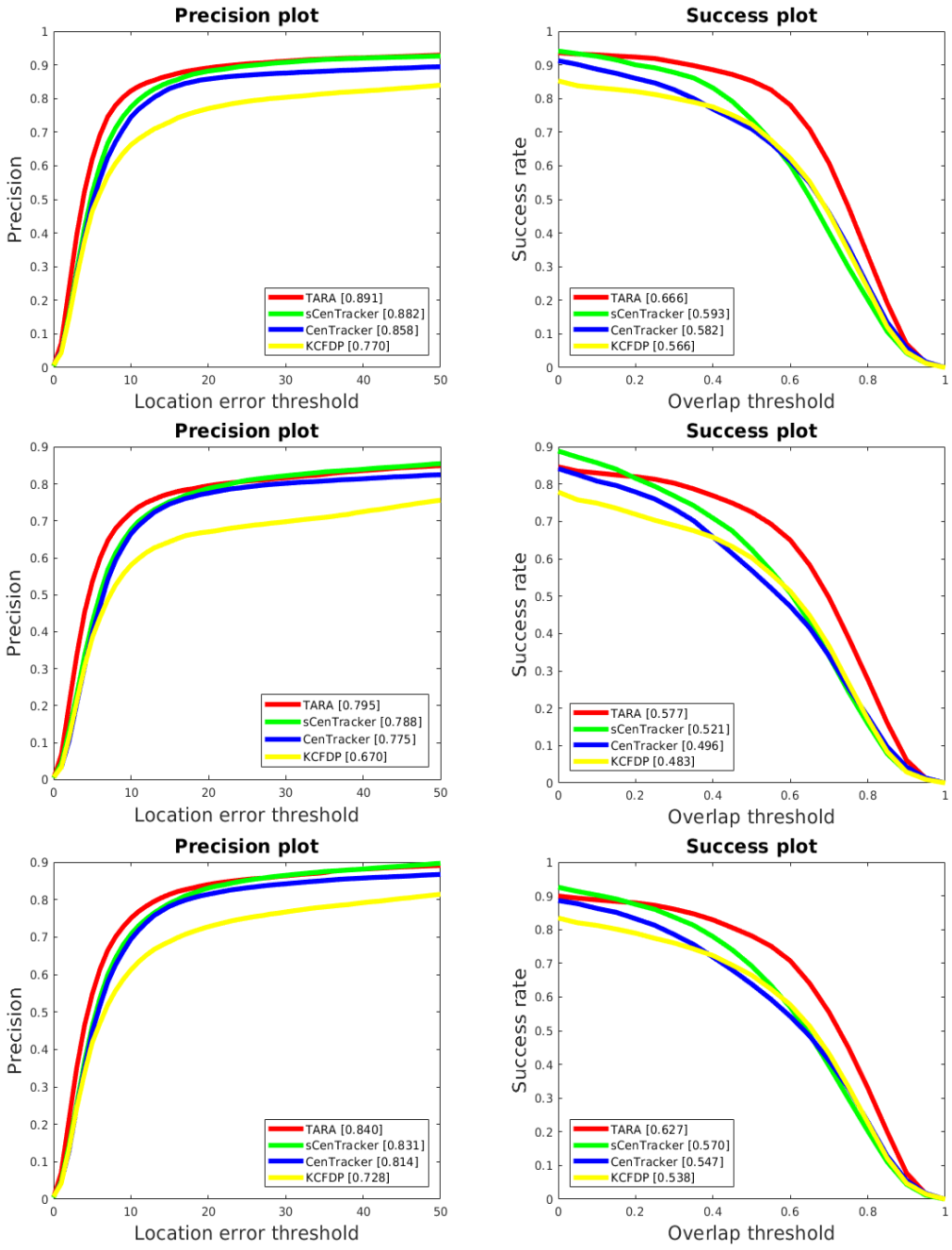

Figure 4.4: The precision plots and success plots for the overall performance of TARA and three representative scale adaptive discriminative correlation filter based trackers on OTB datasets. From top to bottom, the plots correspond to OTB2013, OTB50 and OTB2015, respectively. The left column corresponds to the precision plots, and the right column corresponds to the success plots.

In comparison with KCFDP that employs hand-crafted features, as shown in Table 4.1 and Figure 4.4, the other three trackers (CenTracker, sCenTracker, and TARA) improves the tracking performance by a large margin, which verifies the effectiveness of the CNN features. In comparison with sCenTracker, which obtains the second best performance, our proposed tracker TARA improves the precision score by 0.9%, 0.7% and 0.9% and the AUC score by 7.3%, 5.6% and 5.7% on OTB2013, OTB50, and OTB2015, respectively. The significant performance improvement on AUC score clearly demonstrates the effectiveness of our proposed scale searching scheme. Despite the superior performance, similar to other CNN based trackers, TARA cannot achieve frame-rate operation due to the computation of deep features, and hence is not suitable for real-time applications (*e.g.,* path tracking using unmanned aerial vehicles).

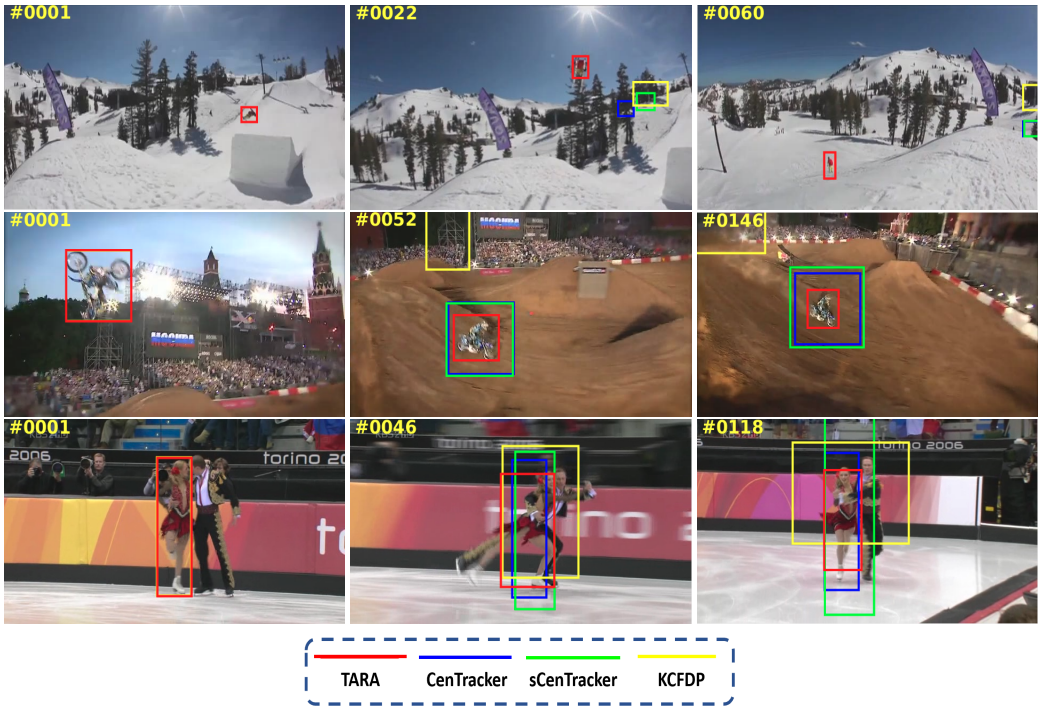

Figure 4.5: Qualitative comparisons between our proposed tracker TARA with three scale adaptive variants of the standard discriminative correlation filter based trackers. Example frames shown are from *Skiing* (top row), *MotorRolling* (second row), and *Skating2.1* (bottom row) sequences that are selected from online tracking benchmark datasets involving both scale variation and aspect ratio variation challenges.

Attribute-Based Comparison

Since TARA is specifically designed for scale estimation, we perform experiments on SV and ARV attributes of the OTB datasets. As shown in Figure 4.5, with aspect ratio adaptability, our tracker significantly increases the robustness and accuracy by estimating the target size accurately and providing a more accurate update of the target appearance model.

To validate the scale adaptability of TARA, experiments are conducted on the SV attribute of OTB datasets and the performance is summarized in Table 4.2. The precision plots and success plots are presented in Figure 4.6.

As shown in Table 4.2 and Figure 4.6, TARA and sCenTracker both achieve promising precision scores due to the employment of the CNN features. Despite similar performance in localization, in comparision with sCenTracker, TARA improves the AUC score by 8.0%, 5.1%, and 6.0% on OTB2013, OTB50, and OTB2015, respectively. The significant performance improvement shows TARA is superior in scale estimation and can address the scale variation challenges effectively.

Table 4.2: Comparisons between TARA and three discriminative correlation filter based trackers on the scale variation attribute (SV) of OTB2013, OTB50, and OTB100 benchmark datasets. The first and the second best values are highlighted in bold and underlined, respectively.

		OTB2013		**OTB50**		**OTB2015**	
	Method	**Prec.**	**AUC**	**Prec.**	**AUC**	**Prec.**	**AUC**
SV	KCFDP	0.744	0.540	0.640	0.463	0.701	0.510
	CenTracker	<u>0.826</u>	0.495	0.762	0.458	0.779	0.477
	sCenTracker	**0.871**	<u>0.567</u>	**0.787**	<u>0.518</u>	<u>0.805</u>	<u>0.541</u>
	TARA	**0.871**	**0.647**	<u>0.782</u>	**0.569**	**0.814**	**0.601**

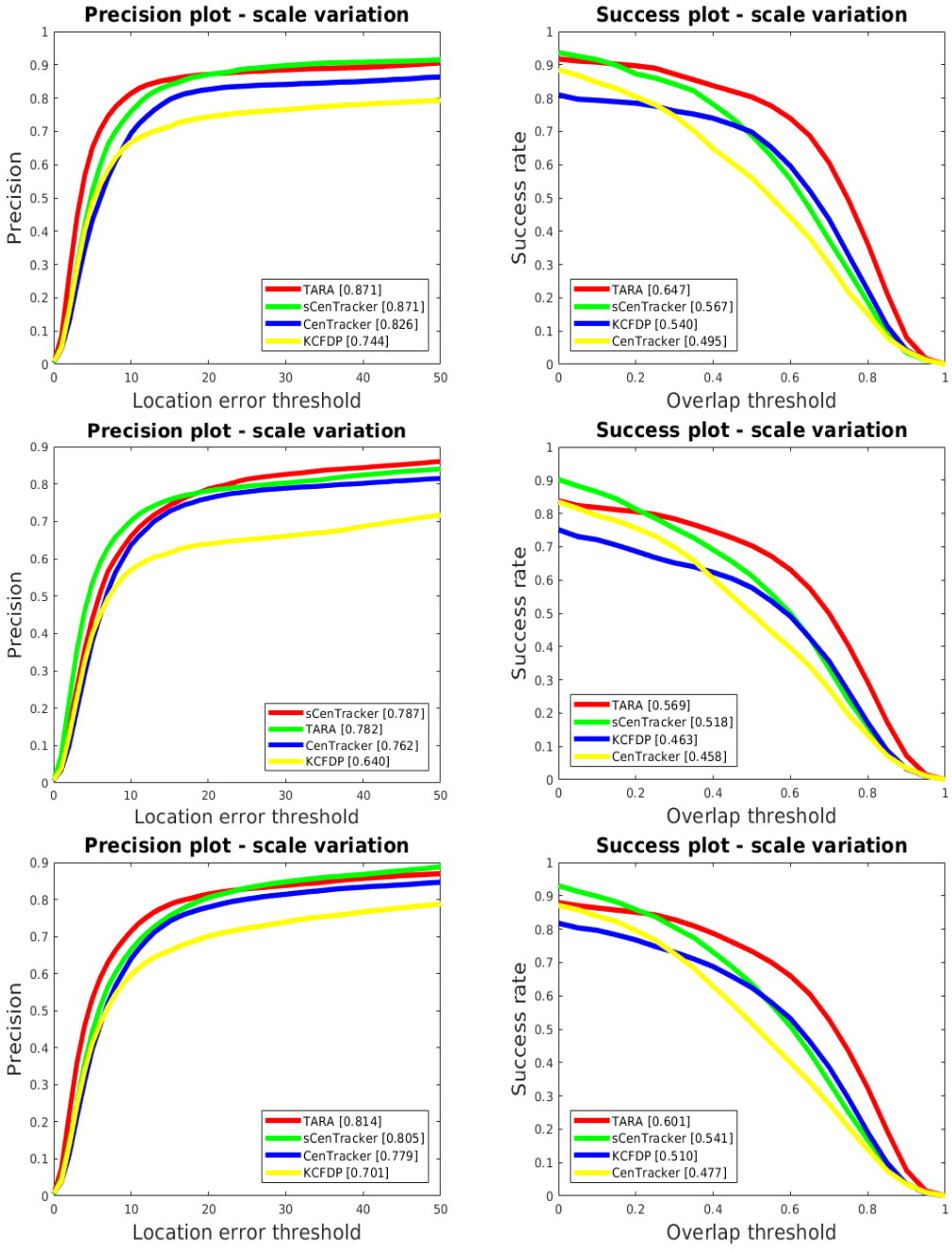

Figure 4.6: The precision plots and success plots of TARA and three representative scale adaptive discriminative correlation filter based trackers on the scale variation attribute of OTB datasets. From top to bottom, the plots correspond to OTB2013, OTB50 and OTB2015, respectively. The left column corresponds to the precision plots, and the right column corresponds to the success plots.

To validate the aspect ratio adaptability of TARA, experiments are performed on the ARV attribute of OTB datasets and the performance is summarized in Table 4.3. The precision and success plots are presented in Figure 4.7.

As shown in Table 4.3 and Figure 4.7, TARA obtains the best performance on all aspect ratio variation attributes of the OTB datasets. Compared to sCenTracker that obtains the second best performance, TARA improves the precision score by 0.1%, 1.1%, and 1.2% and the AUC score by 5.1%, 4.7%, and 4.4% on OTB2013, OTB50, and OTB2015, respectively. The remarkable performance improvement on the aspect ratio variation attribute shows TARA can address the aspect ratio variation challenges effectively compared to the multi-scale searching scheme.

Table 4.3: Comparisons between TARA and three scale adaptive discriminative correlation filter based trackers on the aspect ratio variation (ARV) attribute of OTB2013, OTB50, and OTB100 benchmark datasets. The first and the second best values are highlighted in bold and underlined, respectively.

		OTB2013		**OTB50**		**OTB2015**	
	Method	**Prec.**	**AUC**	**Prec.**	**AUC**	**Prec.**	**AUC**
ARV	KCFDP	0.636	0.432	0.545	0.397	0.609	0.449
	CenTracker	0.781	0.443	0.696	0.417	0.735	0.450
	sCenTracker	<u>0.811</u>	<u>0.493</u>	<u>0.725</u>	<u>0.461</u>	<u>0.758</u>	<u>0.491</u>
	TARA	**0.812**	**0.544**	**0.736**	**0.508**	**0.770**	**0.535**

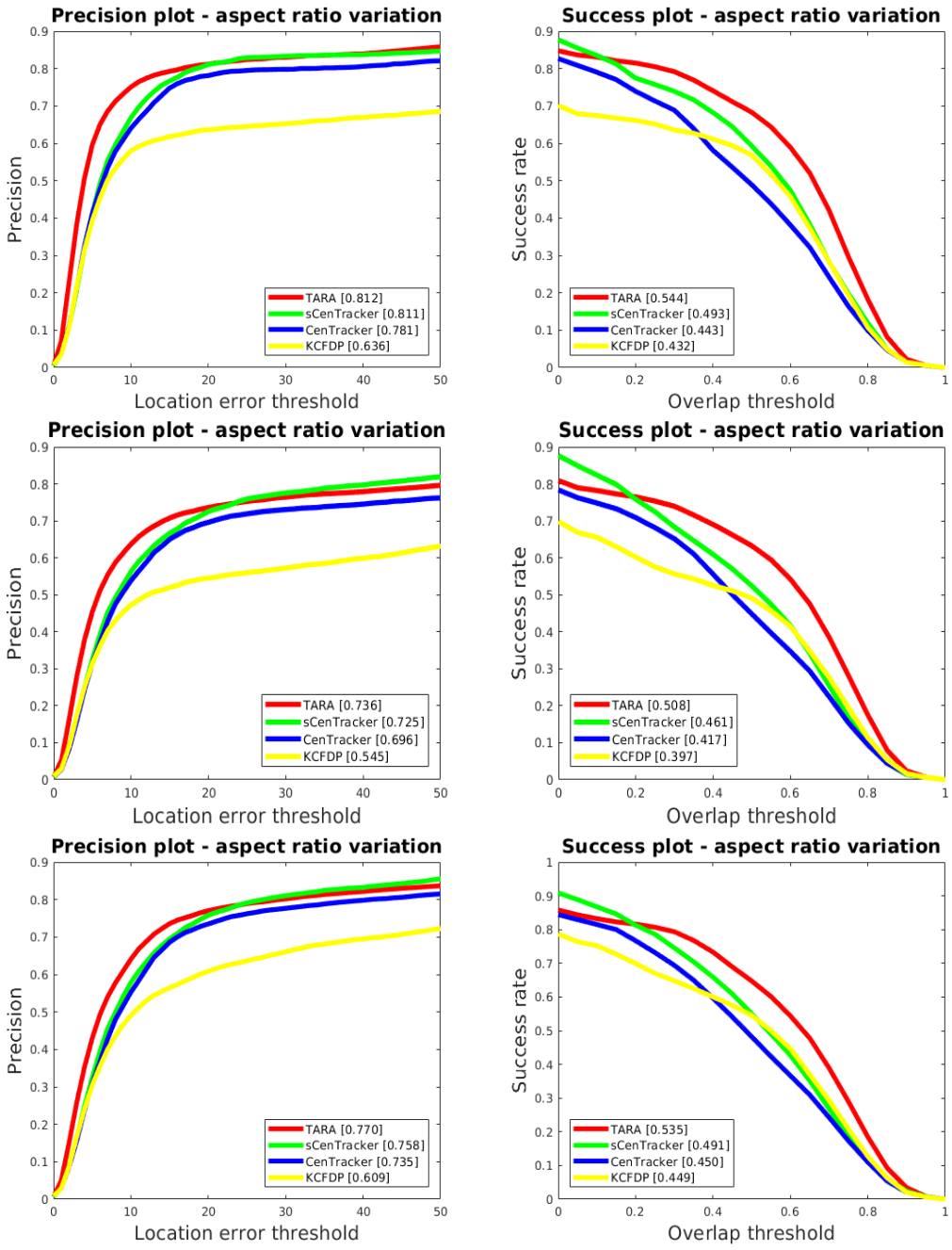

Figure 4.7: The precision and success plots of TARA and three representative scale adaptive discriminative correlation filter based trackers on the aspect ratio variation attribute of the online tracking benchmark datasets. From top to bottom, the plots correspond to OTB2013, OTB50 and OTB2015, respectively. The left column corresponds to the precision plots and the right column corresponds to the success plots.

4.3.4 Internal Analysis of the Proposed Approach

To expose the contributions of different components of TARA, we propose two variants of TARA: TARA without the center 2D DCF (TARA_wo_c) which only relies on four 1D DCFs, TARA without four boundary 1D DCFs (TARA_wo_b) which only relies on the center 2D DCF. For the sake of simplicity, TARA and all two variants are only evaluated on OTB2015 and the performance is summarized in Table 4.4.

Table 4.4: Overall comparisons between TARA and its two variants on the OTB2015 datasets. The first and the second best values are highlighted in bold and underlined.

	TARA_wo_c	TARA_wo_b	TARA
Prec.	0.456	<u>0.814</u>	**0.840**
AUC	0.345	<u>0.547</u>	**0.627**

As shown in Table 4.4, TARA_wo_c provides inferior performance compared to TARA_wo_b, which is reasonable since the boundary DCFs contain less context appearance information than the center DCF and are likely to drift earlier. Compared to TARA_wo_c and TARA_wo_b, TARA improves the precision score by 38.4% and 2.6%, respectively, and the AUC score by 28.2% and 8.0%, respectively. The significant performance improvement shows that the center DCF and the boundary DCFs are complementary. These results validate the effectiveness of key parts of the proposed TARA tracker.

4.4 Discussion

In this chapter, TARA, a scale estimation approach with aspect ratio adaptability, is proposed by localizing the boundaries of the target object using a group of discriminative correlation filters. To accurately localize the target boundary, rich deep hierarchical convolutional features, which encode

target appearance information with different levels of abstraction are exploited. Comprehensive evaluations on the online tracking benchmark datasets validate that TARA can meet the demand of aspect ratio variation challenges. Despite the superior tracking performance, TARA cannot achieve frame-rate operation due to the employment of the deep features in the discriminative correlation filter. Besides, the deep features are computed using the network pretrained for the object detection task, which is not tailored for the tracking task. Therefore, we propose to design a tracker that operates at a frame-rate beyond real-time and train a network end-to-end to exploit the expressive power of deep convolutional networks for tracking, which will be detailed in the next chapter.

5

CAT: Centerness-Aware Anchor-Free Tracker

In order to obtain a scale estimation of the target object, most methods rely either on a multi-scale searching scheme or on refining a set of predefined anchor boxes. These methods require heuristically selected parameters, such as scale factors of the multi-scale searching scheme as in references 36 and 42, or sizes and aspect ratios of the predefined candidate anchor boxes as in reference 51.

Besides, as shown in reference 52, the performance is sensitive to the aforementioned predefined hyperparameters. On the contrary, a centerness-aware anchor-free tracker (CAT) is introduced in this chapter. First, the location and scale of the target object are predicted in an anchor-free fashion by decomposing tracking into parallel classification and regression problems. The proposed anchor-free design obviates the need for hyperparameters related to the anchor boxes, making CAT more generic and flexible. Second, with the observation that many low-quality predictions are produced corresponding to the locations far off the target center, the proposed centerness-aware classification branch is designed to identify the foreground from the background while predicting the normalized distance from the location within the foreground to the target center, *i.e.*, the centerness. The output of the proposed centerness-aware classification branch is employed to suppress low-quality state estimates to improve the tracking performance. The experiments show that our centerness-aware anchor-free tracker, with its appealing features, achieves salient performance in a wide variety of tracking scenarios.

The organization of this chapter is described as follows. In Section 5.1, the background and motivation of the proposed tracker CAT are given. In Section 5.2, Siamese network based trackers and anchor-free mechanisms that are most relevant to our approach are reviewed. In Section 5.3, the details of our proposed tracker CAT are presented, and the main components of the proposed network including the feature extraction, the feature combination, and the state estimation components are described in Section 5.3.1, Section 5.3.2 and Section 5.3.3, respectively. The details about the loss function are described in Section 5.3.4. The relationship between prior anchor-free methods and our proposed tracker are detailed in Section 5.3.5. In Section 5.4.1, the implementation details of the training and inference stages are described. The benchmark datasets and evaluation metrics are detailed in Section 5.4.2. In Section 5.4.3, the qualitative and quantitative comparison results are presented. In Section 5.4.4, an ablation test is performed to reveal the contribution of each component. Section 5.5 concludes the chapter.

5.1 Background

Recently, visual trackers based on a Siamese network[21,51,53,54] have received considerable attention due to the balanced accuracy and speed. In the pioneering work, a fully convolutional Siamese network (SiamFC)[21] is achieved via training a Siamese network to perform template matching and compute a similarity score between the exemplar image of the target object and the search image. In order to adjust to scale variation, a multi-scale searching scheme is employed as in references 36 and 42 to perform template matching on multiple scales to identify the best scale. However, the multi-scale searching scheme is computationally expensive when facing large scale variations due to the multi-scale feature extraction step. To address this issue, the Siamese region proposal network (SiamRPN)[53] employs a regional proposal network (RPN)[55], which consists of a classification network for foreground-background estimation and a regression network for the predefined candidate anchor boxes refinement, *i.e.,* learning two-dimensional offsets to the predefined anchor boxes. This representative anchor-based method avoids the time-consuming multi-scale feature extraction step in the multi-scale searching scheme while achieving promising accuracy. However, as shown in reference 52, the hyperparameters related to the anchor boxes (*e.g.*, numbers, sizes, and aspect ratios) entail tedious and heuristic tuning and have a large impact on the final performance.

Here, we design a centerness-aware anchor-free tracker by decomposing the tracking into parallel classification and regression problems. With the above decomposition, the tracking problem is addressed in a per-pixel fashion. The regression branch regresses each spatial location a relative bounding box by predicting the distance from the corresponding spatial location to each side of the axis-aligned ground truth bounding box. Based on the observation that many low quality predictions generated correspond to the locations that are far away from the center of the target object, the classification branch is designed to learn to output a 0 for the background and a value ranging from 0 to 1 to indicate the normalized distance between the spatial location within the foreground and

the target center, thus suppressing low quality state estimations.

The main contributions in this chapter can be summarized as follows. First, an effective anchor-free tracking technique is proposed, which avoids the heuristic tuning related to the anchor boxes during training. Second, the proposed classification branch is centerness-aware and can improve the tracking accuracy and robustness significantly by suppressing low-quality state estimates. Third, proof of validation and the efficacy of CAT is achieved through extensive experiments that are performed on publicly available benchmark datasets.

5.2 Related Work

In this section, we review visual trackers based on the Siamese network and anchor-free mechanism that are closely related to our proposed tracker CAT.

5.2.1 Siamese Network Based Trackers

In recent years, the Siamese network has been extensively used in the tracking community because of its balanced accuracy and efficiency. In the pioneering work of SiamFC [21], the Siamese network is employed for feature extraction, and a cross-correlation layer is used to calculate a similarity score between the target and candidate image patches. Furthermore, a transformation learning model is proposed in the dynamic Siamese network (DSiam) [56] to learn the appearance change of the target object more effectively and to suppress the background. In the residual attentional Siamese network (RASNet) [57], various attention mechanisms are incorporated for better adaption to the current target state. The Siamese network with semantic and appearance branches (SA-Siam) [58] achieves improved performance through training the two branches separately so that the heterogeneity of different types of features can be retained. Although promising results have been achieved, these methods resort to the multi-scale searching scheme, and cannot accommodate large scale variations

and aspect ratio variations.

To tackle the large scale variation and aspect ratio variations through refining the predefined candidate anchor boxes, SiamRPN[53] combines SiamFC[21] with a RPN[55] and obviates the need for the time-consuming step of extracting feature maps at multiple scales. Furthermore, the series-parallel matching real-time tracker (SPM-tracker)[59] improves the robustness and discrimination power of SiamRPN via a series-parallel matching scheme. Through increasing the hard negative training data in the training stage, the distractor-aware Siamese regional proposal network (DaSiamRPN)[54] improves the robustness of the model. Further, a multi-stage tracking framework C-RPN[51] is proposed by cascading a sequence of RPNs to fully explore features across layers for robust visual tracking. However, as shown in reference 52, the hyperparameters related to the anchor boxes require heuristic tuning to achieve good performance. In contrast, CAT addresses VOT in an anchor-free manner and obviates the tuning related to the hyperparameters of the anchor boxes and hence is more flexible.

5.2.2 Anchor-Free Mechanism

Recently, anchor-free methods have been widely employed in a family of object detectors due to their simple architectures and promising performance. In contrast to the anchor-based methods that refine the predefined candidate anchor boxes, the anchor-free methods estimate the target state in a more direct way. Generally, the anchor-free detectors can be grouped into keypoint-based object detectors and dense object detectors.

In Dense RepPoints[60], the object is represented as a large set of keypoints to describe the object at multiple levels. On the other hand, in CornerNet[61], the target state is described as paired keypoints and detected using a single convolution neural network. Similar to CornerNet, the target state is represented as four extreme points (top-most, left-most, bottom-most, right-most), together with one center point in ExtremeNet[62], and detected using a standard keypoint estimation net-

work. Based on CornerNet, CenterNet[63] proposes to exploit the global information of the object to improve the performance. In reference 64, the object is modeled as a single point (the center), and all other object proprieties (*e.g.*, size and orientation) are then regressed directly from image features corresponding to the center location. In the feature-selective anchor-free object detector (FSAF)[65], a feature-selective module is proposed to address the heuristic-guided feature selection and overlap-based anchor sampling problems that exist in the conventional anchor-based detectors.

In contrast to the keypoint-based methods, FoveaBox in reference 66 learns the object and the bounding box coordinates without any references (*e.g.*, anchors or keypoints). Furthermore, the fully convolutional one-stage object detector FCOS[67] treats the object detection as a binary foreground-background classification for each spatial location, together with the regression of the relative bounding box for the corresponding location. In addition, the centerness branch is proposed to suppress low quality predictions corresponding to the locations that are far from the target center. In VarifocalNet[68], a new loss function and a new star-shaped bounding box feature representation are proposed to improve the ranking and refinement of the bounding boxes. In tracking, inspired by FCOS, SiamCAR is proposed in reference 69 to address the tracking problem by decomposing the tracking task into one classification problem, one centerness estimation problem, and one regression problem.

5.3 Proposed Method

In this section, the details of our proposed tracker CAT are described. As shown in Figure 5.1, CAT is built upon three main components: a Siamese network as a feature extractor, a convolution layer with a kernel size of 1×1 after a depth-wise cross-correlation operation for feature combination, and a centerness-aware anchor-free network for state estimation.

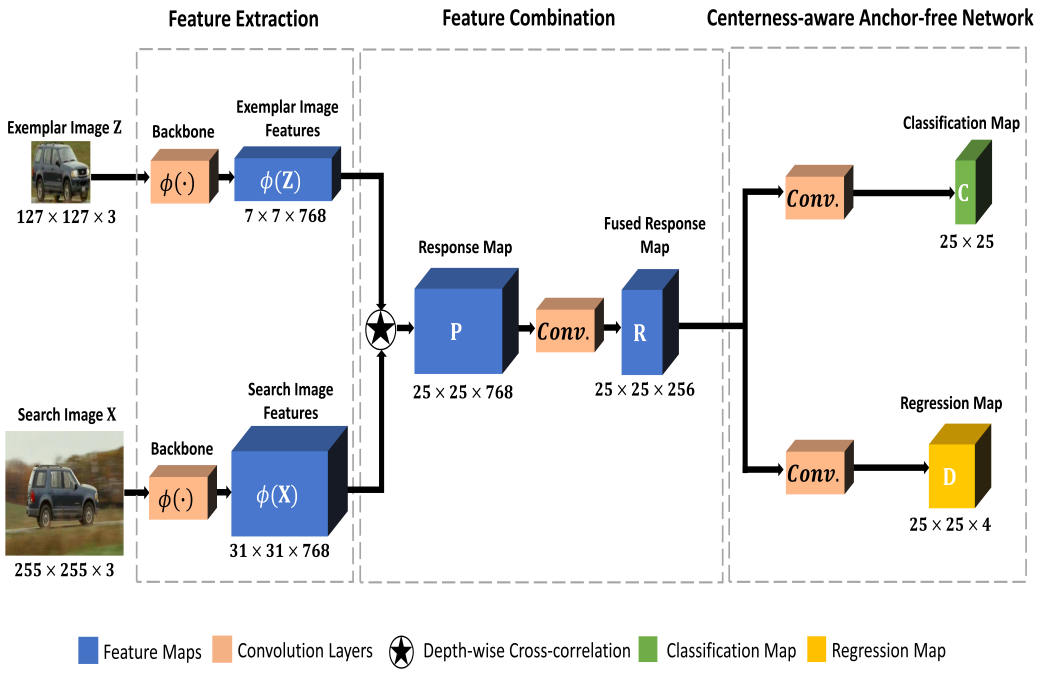

Figure 5.1: The tracking pipeline of the proposed CAT tracker. The backbone Siamese network takes exemplar image $\mathbf{Z}$ and search image $\mathbf{X}$ as input and outputs corresponding feature maps denoted as $\phi(\mathbf{Z})$ and $\phi(\mathbf{X})$. In order to embed the features from two branches, a depth-wise cross-correlation operation is employed to obtain the multi-channel response map denoted as $\mathbf{P}$. Then, to reduce the computational cost, a convolution layer with a kernel size of 1×1 is employed to fuse the response map. The fused response map with reduced dimension is denoted as $\mathbf{R}$ and is adopted as the input to the centerness-aware anchor-free network. Regarding every spatial location on the regression map $\mathbf{D}$, the regression branch learns to estimate the distance from the corresponding location to each side of the ground truth bounding box. For the classification map $\mathbf{C}$, with the observation that many low-quality predictions are produced corresponding to the locations far from the target center, the centerness-aware classification branch learns to output a 0 for the background and a value ranging from 0 to 1 to indicate the normalized distance between the spatial location within the foreground and the target center to suppress predictions with low quality.

5.3.1 Feature Extraction

The Siamese network takes an exemplar image and a search image as input. The image patch with the center point as the target center is cropped in the initial frame, and is then adopted as the exemplar image. The search images are cropped in the following frames to denote the search region. The

Siamese network includes two branches that share the same weights to embed both patches into the same feature space. The Siamese network takes the exemplar image $\mathbf{Z}$ and the search image $\mathbf{X}$ as input and outputs $\phi(\mathbf{Z})$ and $\phi(\mathbf{X})$ as the corresponding feature maps.

In our proposed tracker, A modified ResNet-50[70] is used for feature extraction. Specifically, the strides of the last two residual blocks are reduced to eight, and thus the last three residual blocks output feature maps with the same spatial resolution. Moreover, an extra 1×1 convolution layer is appended to each of the block outputs to reduce the channel number to 256. We aggregate hierarchical features as in reference 71 to achieve higher robustness and accuracy. Formally, the features generated from the last three residual blocks are respectively denoted as $f_3(\cdot)$, $f_4(\cdot)$, and $f_5(\cdot)$ and combined as

$$\phi(\cdot) = f_3(\cdot) \mathbin{+\!\!+} f_4(\cdot) \mathbin{+\!\!+} f_5(\cdot), \tag{5.1}$$

where $+\!\!+$ is the concatenation operation; $f_3(\cdot)$, $f_4(\cdot)$, and $f_5(\cdot)$ each include 256 channels; and $\phi(\cdot)$ consists of 768 channels.

5.3.2 Feature Combination

A depth-wise cross-correlation layer is employed here to embed the features from the two branches and generate the multi-channel response map $\mathbf{P}$ as

$$\mathbf{P}_k = \phi_k(\mathbf{Z}) \star \phi_k(\mathbf{X}), \tag{5.2}$$

where $\star$ represents the cross-correlation operation, and $\phi_k(\mathbf{Z})$, $\phi_k(\mathbf{X})$, and $\mathbf{P}_k$ denote the k-th channel of $\phi(\mathbf{Z})$, $\phi(\mathbf{X})$, and $\mathbf{P}$, respectively. Furthermore, a 1×1 convolution layer is employed to fuse the response map and reduce the dimension from 768 channels to 256 channels. Through the dimension reduction operation, the number of parameters can be reduced to speed up the following

computation. The response map with the reduced dimension is denoted as $\mathbf{R}$ and employed as the input to the centerness-aware anchor-free network.

5.3.3 Centerness-Aware Anchor-Free Network

The centerness-aware anchor-free network includes two branches: a regression branch and a classification branch that is centerness aware. As in Figure 5.1, regarding the input $\mathbf{R}$, the centerness-aware classification branch produces the classification map $\mathbf{C}_{w\times h}$, and the regression branch produces the regression map $\mathbf{D}_{w\times h\times 4}$, where w and h denote the width and height of the classification and regression maps. Each location (m, n) on $\mathbf{C}_{w\times h}$ or $\mathbf{D}_{w\times h\times 4}$ corresponds to one spatial location on the search image $\mathbf{X}$ as $(p_m, p_n) = (\lfloor \frac{s}{2} \rfloor + m \times s, \lfloor \frac{s}{2} \rfloor + n \times s)$, where s denotes the network stride and $\lfloor \cdot \rfloor$ represents the floor operation. Compared to the anchor-based methods that take every location as the center point of a set of predefined anchor boxes and predict the offset values regarding the anchor boxes, we predict the target bounding box for that spatial location directly.

The regression branch learns to output the distance from the spatial location to each side of the ground truth bounding box. Let the width, height, center point, upper-left corner, and lower-right corner be g_w, g_h, (g_{x_c}, g_{y_c}), $(g_{x_{ul}}, g_{y_{ul}})$, and $(g_{x_{lr}}, g_{y_{lr}})$, respectively. The regression targets $\mathbf{D}^t_{m,n} = (l_{m,n}, t_{m,n}, r_{m,n}, b_{m,n})$ for location (m, n) are computed as

$$\begin{aligned} l_{m,n} &= p_m - g_{x_{ul}}, \\ t_{m,n} &= p_n - g_{y_{ul}}, \\ r_{m,n} &= g_{x_{lr}} - p_m, \\ b_{m,n} &= g_{y_{lr}} - p_n, \end{aligned} \tag{5.3}$$

where $l_{m,n}$, $t_{m,n}$, $r_{m,n}$, and $b_{m,n}$ denote the distance from (p_m, p_n) to the left side, top side, right side, and bottom side of the ground truth bounding box, respectively.

Different from the centerness branch in FCOS[67] and SiamCAR[69] that only learns to output centerness value for the foreground spatial locations and generate uncontrollable values for background spatial locations, our proposed centerness-aware classification branch acts as the joint representation of the binary classification and the centerness branches that learns to differentiate the foreground from the background and to output the centerness value for the foreground spatial locations to denote the normalized distance between the spatial location within the foreground and the target center. Specifically, the target value $\mathbf{C}^t_{m,n}$ for location (m, n) is set to be 0 for the background and ranges from 0 to 1 for the foreground to denote the normalized distance from location (p_m, p_n) to the target center. Formally, $\mathbf{C}^t_{m,n}$ is defined as

$$\mathbf{C}^t_{m,n} = \mathbb{I}_c(\mathbf{D}^t_{m,n}) \times \sqrt{\frac{\min(l_{m,n}, r_{m,n})}{\max(l_{m,n}, r_{m,n})} \times \frac{\min(t_{m,n}, b_{m,n})}{\max(t_{m,n}, b_{m,n})}}, \tag{5.4}$$

where $\mathbb{I}_c$ is the indicator function defined as

$$\mathbb{I}_c(\mathbf{D}^t_{m,n}) = \begin{cases} 1, & \text{if } \min(\mathbf{D}^t_{m,n}) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5.5}$$

The regression branch is composed of four convolutional layers with a kernel size of 3×3 and a channel number as 256 and one convolutional layer with the same kernel size and a channel number of 4. Then, an exponential operation is used to map the real number output into $(0, +\infty)$ because the target values of the regression branch output are positive real numbers. The centerness-aware classification branch has the same structure with the only difference being that the channel number of the last convolutional layer is 1 instead of 4. Then, the output will be mapped into $(0, 1)$ through a sigmoid function.

5.3.4 Loss Function

We adopt the intersection over union (IoU) loss[72] for the regression branch. Here, we only treat the locations within an elliptical area centered within the ground truth bounding box as positive samples as in reference 67. More concretely, the regression loss $\mathcal{L}_{\text{reg}}$ is defined as

$$\mathcal{L}_{\text{reg}} = \frac{-1}{\sum_{m,n} \mathbb{I}_d(\mathbf{D}^t_{m,n})} \sum_{m,n} \mathbb{I}_d(\mathbf{D}^t_{m,n}) \times \ln\left(\text{IoU}(\mathbf{GT}, \mathbf{B}_{m,n})\right), \tag{5.6}$$

where $\mathbf{GT}$ and $\mathbf{B}_{m,n}$ denote the ground truth bounding box and the predicted bounding box corresponding to location (m, n), and $\mathbb{I}_d(\mathbf{D}^t_{m,n})$ is the indicator function defined as

$$\mathbb{I}_d(\mathbf{D}^t_{m,n}) = \begin{cases} 1, & \text{if } \left(\frac{p_m - g_{x_c}}{\alpha \times g_w}\right)^2 + \left(\frac{p_n - g_{y_c}}{\alpha \times g_h}\right)^2 < 1, \\ 0, & \text{otherwise,} \end{cases} \tag{5.7}$$

with α controlling the size of the area corresponding to the positive samples in the training of the regression branch.

Acting as a joint representation of the binary classification and centerness estimation branches, the proposed centerness-aware classification branch does pose additional challenges to the training process. The training of the proposed centerness-aware classification branch needs dense supervisions across the whole search image and hence may result in a data imbalance problem. Moreover, compared to a binary classification that only contains discrete labels, the continuous labels ranging from 0 to 1 make the training more challenging. As shown in Section 5.4.4, the standard cross-entropy loss cannot successfully train the proposed centerness-aware classification branch due to challenges in the training process. Therefore, to ensure the successful training, we combine the focal

loss[52] with the cross-entropy loss[73] as

$$\begin{aligned}\mathcal{L}_c = & \frac{-1}{\sum_{m,n} \mathbb{I}_d(\mathbf{D}^t_{m,n})} \sum_{m,n} \left|\mathbf{C}_{m,n} - \mathbf{C}^t_{m,n}\right|^{\beta} \\ & \times \left(\mathbf{C}^t_{m,n} \times \ln\left(\mathbf{C}_{m,n}\right) + (1 - \mathbf{C}^t_{m,n}) \times \ln\left(1 - \mathbf{C}_{m,n}\right)\right),\end{aligned} \tag{5.8}$$

where $-\left(\mathbf{C}^t_{m,n} \times \ln\left(\mathbf{C}_{m,n}\right) + (1 - \mathbf{C}^t_{m,n}) \times \ln\left(1 - \mathbf{C}_{m,n}\right)\right)$ acts as the cross-entropy part. Similar to reference 52, the modulating factor $\left|\mathbf{C}_{m,n} - \mathbf{C}^t_{m,n}\right|^{\beta}$ reflects the difference between the estimation $\mathbf{C}_{m,n}$ and the continuous label $\mathbf{C}^t_{m,n}$ and makes the training process focus more on the hard samples via reducing the loss contribution of the easy samples, and β balances the importance of easy and hard samples. Hence, the training of the entire network is to minimize the multi-task loss defined as

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_c + \lambda_2 \mathcal{L}_{\text{reg}}, \tag{5.9}$$

where λ_1 and λ_2 are the tradeoff parameters that balance the importance of $\mathcal{L}_c$ and $\mathcal{L}_{\text{reg}}$.

5.3.5 Relationship to Prior Anchor-Free Work

CAT shares some motivating factors with the recent detection method FCOS[67] and the recent tracking method SiamCAR[69]. Here, we discuss the differences regarding these two related works.

In FCOS, the task is to detect the category of the objects, and the categories are predefined. In contrast, in VOT, it is necessary to determine the object at the instance level. Hence, in CAT, a template branch is adopted to encode the target object appearance information. Both FCOS and SiamCAR use a binary classification branch to identify foreground from background and a branch to predict the centerness value to indicate the normalized distance from the corresponding spatial location to the target center. The low-quality predictions that are far from the target center are suppressed via multiplying the outputs of the centerness branch with those of the binary classification

branch. However, it should be noted that, in both the FCOS and SiamCAR methods, the centerness branch is only trained with respect to the centerness value corresponding to spatial locations within the groundtruth bounding box, as the centerness value is not defined for background spatial locations. Hence, in the centerness branch, the output for the background spatial location is unconstrained. This will not create any trouble in the most ideal scenario, where the binary classification branch is perfectly trained and outputs exact 0 and 1 for background and foreground spatial locations, respectively. However, in the inference stage, the binary classification branch will not output exact 0 and 1 and may result in an inconsistency risk. For example, when the background classification score is not 0 and the foreground classification score is not high enough, it is possible that the state estimation corresponding to background spatial locations is chosen as the tracking result due to the high unconstrained centerness score corresponding to the background spatial locations. In addition, since the regression branch is only trained with respect to the foreground spatial locations, using the output of the regression branch that corresponds to the background spatial locations cannot provide an accurate state estimation of the target object and may result in tracking failures. Based on these observations, we propose the centerness-aware classification branch to directly learn the products of binary classification branch outputs and the centerness branch outputs in SiamCAR to mitigate the inconsistency risk between the training and inference stages. Different from the centerness branch in FCOS[67] and SiamCAR[69], our proposed centerness-aware classification branch will act as the joint representation of the binary classification branch and the centerness estimation branch to output 0 for the background and the centerness value for the foreground, hence mitigating the inconsistency risk and improving the tracking performance with reduced computation cost and fewer associated parameters.

5.4 Experiments

Extensive experiments have been performed on benchmark datasets to verify the efficacy of our proposed tracker. The implementation details about training and inference are presented in Section 5.4.1. Section 5.4.2 describes the employed benchmark datasets and the evaluation protocols. Qualitative and quantitative comparison results are presented in Section 5.4.3. In Section 5.4.4, we perform an ablation test to show the contribution of each component.

5.4.1 Implementation Details

Training

As in reference 21, the exemplar image size is set as 127×127 pixels, and the search image size is set as 255×255 pixels. The training datasets include COCO[74] , YouTube-BB[75], ImageNet VID and DET[76], and GOT-10k[77]. The weights of the backbone Siamese network are set as the pretrained weights on ImageNet. We train the entire network via stochastic gradient descent, in which the minibatch size is set as 64 and the number of epochs is set to 20. The learning rate will be increased from 0.001 to 0.005 using a warm-up strategy in the first 5 epochs, and will be decayed from 0.005 to 0.0005 in the remaining 15 epochs. The weights of the backbone Siamese network is fixed in the first half training stage and fine-tuned with one-tenth of the present learning rate for the remaining epochs. The values of the momentum and the weight decay are set as 0.9 and 10^{-3}, respectively. The parameters α, β, λ_1, and λ_2 are set as 0.6, 2, 2, and 6, respectively. Our approach is implemented in Python using PyTorch on a PC with Intel Core i9-9820X CPU 3.30 GHz, 64GB RAM, two NVIDIA GeForce RTX 2080 Ti GPUs, and one NVIDIA GeForce GTX TITAN V GPU.

Inference

During inference, we adopt the offline tracking strategy as in references 21 and 51. The features of the exemplar image are computed only in the first frame and used to match the subsequent search images. The search image is cropped centered on the previous estimated target center. The predicted bounding box $\mathbf{B}_{m,n}$ for location (m, n) is computed as

$$\begin{aligned} b_{x_{ul}} &= p_m - \mathbf{D}_{m,n,0}, \\ b_{y_{ul}} &= p_n - \mathbf{D}_{m,n,1}, \\ b_{x_{lr}} &= p_m + \mathbf{D}_{m,n,2}, \\ b_{y_{lr}} &= p_n + \mathbf{D}_{m,n,3}, \end{aligned} \tag{5.10}$$

where $(b_{x_{ul}}, b_{y_{ul}})$ and $(b_{x_{lr}}, b_{y_{lr}})$ represent the upper-left corner and the lower-right corner coordinates of the predicted bounding box $\mathbf{B}_{m,n}$. As in reference 51, a scale change penalty and Hanning window are used to suppress extreme scale variation and fast motion and to obtain the final classification map. In the final classification map, the maximum value suggests the location of the target object. To maintain a smooth change of the shape of the bounding box prediction, the estimated size is updated through linear interpolation with the estimated size from the last frame.

5.4.2 Datasets and Evaluation Metrics

As in references 36, 42 and 71, all trackers are quantitatively evaluated on the online tracking benchmark (OTB) datasets (OTB2013, OTB50, and OTB2015), and three standard evaluation metrics are employed including precision score, success rate, and speed. The detailed description of the benchmark datasets and evaluation metrics can be found in Section 2.4.2 and Section 4.3.2.

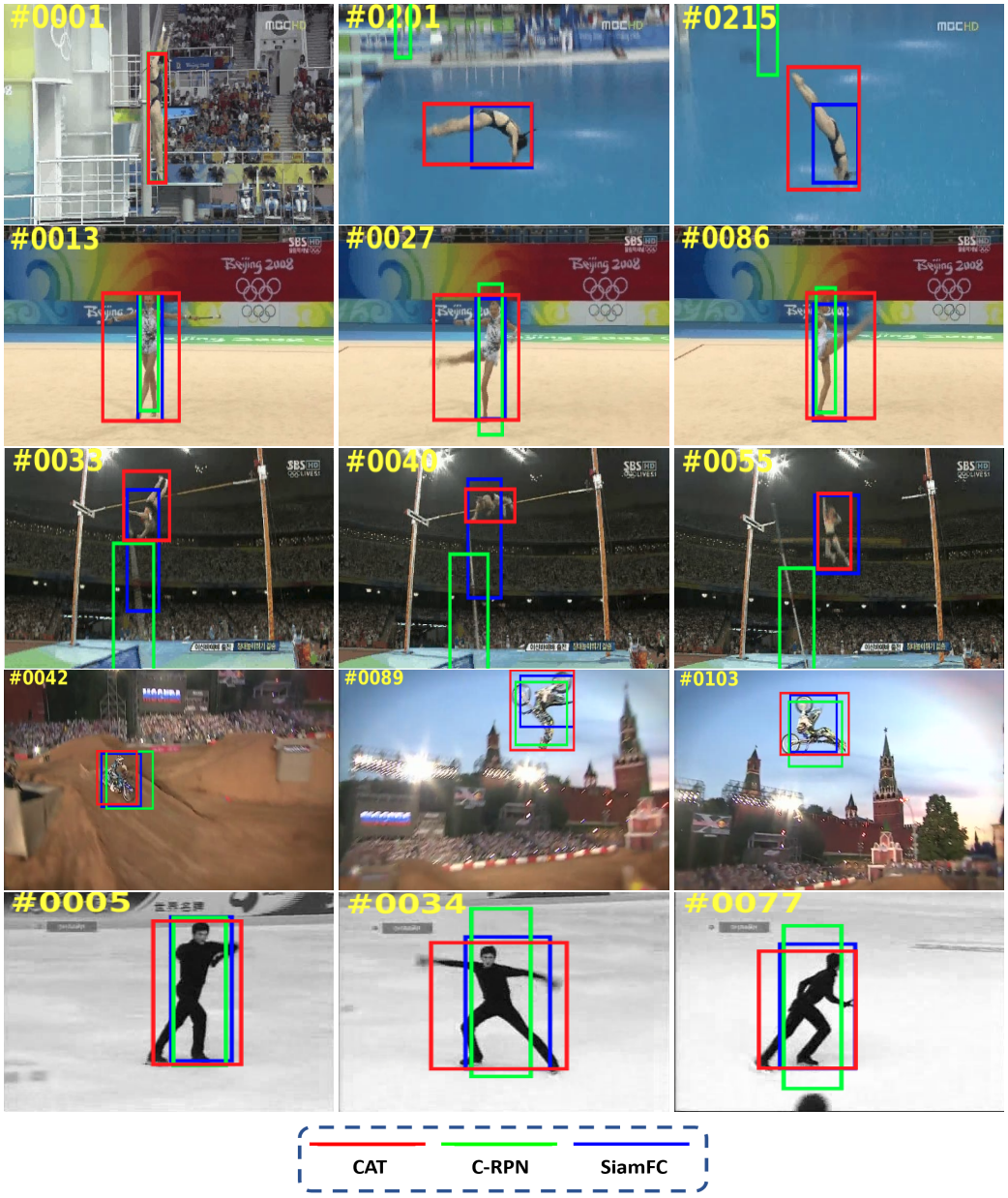

Figure 5.2: Qualitative comparisons between the proposed tracker CAT and two representative Siamese network based trackers SiamFC, and C-RPN on *Diving* (first row), *Gym* (second row), *Jump* (third row), *MotorRolling* (fourth row), and *Skater2* (bottom row) sequences that involve large scale variations and aspect ratio variations. Compared to other trackers, even facing challenging scenarios including occlusion, large scale variations, and aspect ratio variations, CAT provides accurate state estimations that significantly improve the robustness and accuracy in tracking.

5.4.3 Performance Comparison

Overall Comparison

To validate the superior performance of the proposed tracker, CAT is compared to two representative Siamese network based trackers including SiamFC[21] that employs the multi-scale searching scheme to perform template matching on multiple scales to identify the best scale, and C-RPN[54] that employs an anchor-based method for state estimation.

All trackers are evaluated on OTB2013, OTB50, and OTB2015 datasets. Table 5.1, Figure 5.2 and Figure 5.3 summarize the corresponding results. As shown in Figure 5.2, even when facing partial occlusion, large scale variations, and aspect ratio variations, our tracker provides accurate state estimation, which validates the robustness and accuracy of our approach. As in Table 5.1, our proposed tracker CAT obtains the best performance in terms of both AUC score and precision score, and SiamFC has a large performance gap compared to the CAT despite the highest running speed. Compared to C-RPN, CAT improves the precision score by 1.4%, 3.7%, and 3.6% and the AUC score by 1.3%, 3.2%, and 3.4% on OTB2013, OTB50, and OTB2015, respectively. These results are indicative of the efficacy and robustness of the proposed anchor-free centerness-aware tracking scheme.

Table 5.1: Overall comparisons between CAT and two scale adaptive Siamese network based trackers on the online tracking benchmark datasets. The highest and the second highest values are highlighted in bold and underlined, respectively.

		OTB2013		**OTB50**		**OTB2015**	
Method	**Speed**	**Prec.**	**AUC**	**Prec.**	**AUC**	**Prec.**	**AUC**
SiamFC	**46.2**	0.794	0.611	0.683	0.517	0.763	0.585
C-RPN	31.7	<u>0.871</u>	<u>0.647</u>	<u>0.831</u>	<u>0.609</u>	<u>0.846</u>	<u>0.636</u>
CAT	<u>35.6</u>	**0.885**	**0.670**	**0.868**	**0.641**	**0.882**	**0.670**

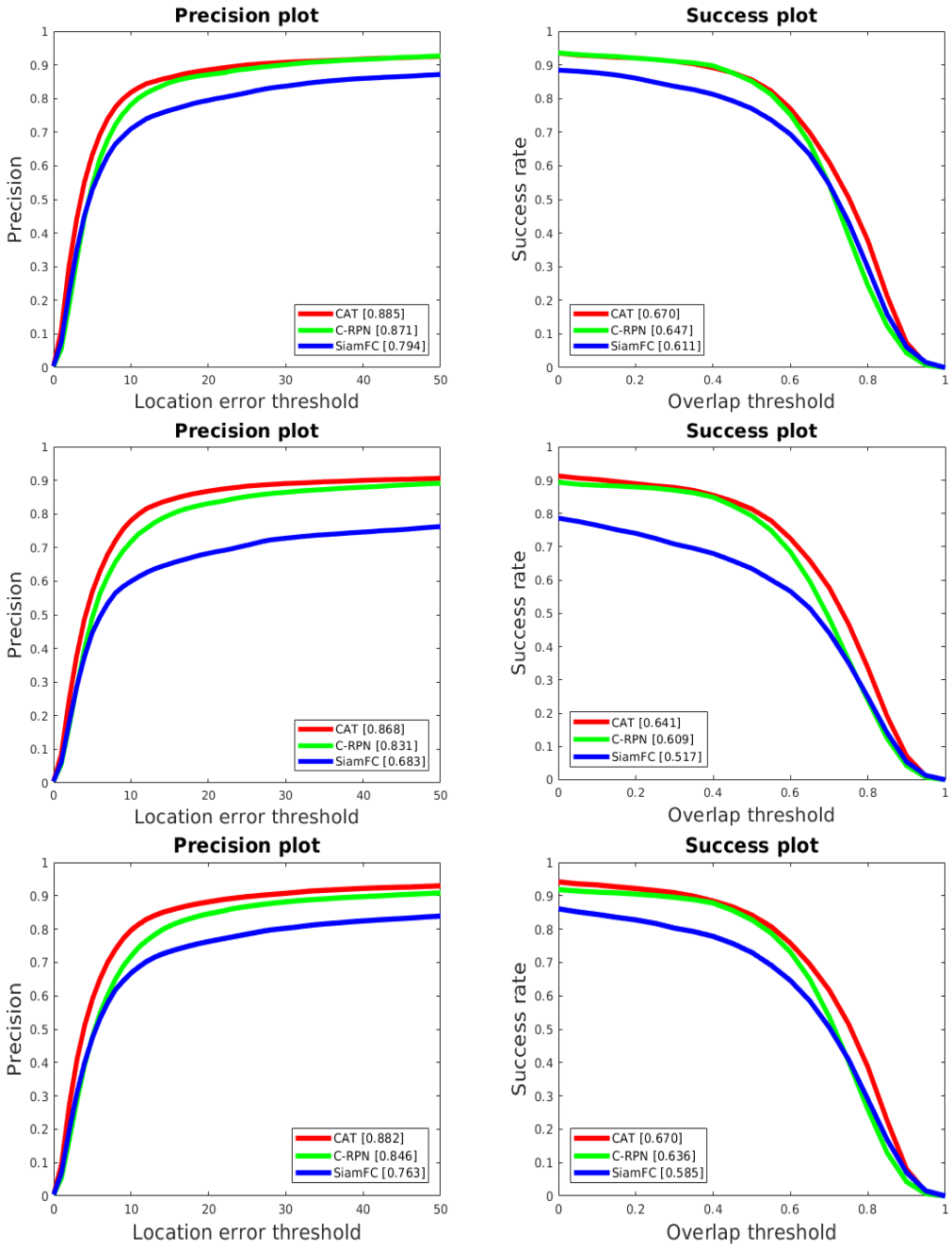

Figure 5.3: The precision plots and success plots for the overall performance of CAT and two representative scale adaptive Siamese network based trackers on the online tracking benchmark datasets. From top to bottom, the plots correspond to OTB2013, OTB50 and OTB2015, respectively. The left column corresponds to the precision plots, and the right column corresponds to the success plots.

Attribute-Based Comparison

To verify the effectiveness of our proposed tracker when dealing with scale variation challenges, we perform experiments on the scale variation attributes of the OTB datasets. The performance is summarized in Table 5.2. The precision plots and success plots for the scale variation attribute are presented in Figure 5.4.

As shown in Table 5.2 and Figure 5.4, with aspect ratio adaptability, our proposed tracker CAT significantly increases the robustness and accuracy by estimating the target size accurately. In comparison with C-RPN, which obtains the second best performance, CAT improves the precision score by 2.5%, 6.0%, and 4.0% and the AUC score by 4.1%, 5.4%, 4.0% on OTB2013, OTB50, and OTB2015, respectively. The promising improvement verifies the effectiveness of our proposed centerness-aware anchor-free tracking mechanism in dealing with the scale variation challenges.

Table 5.2: Overall comparisons between CAT and two representative scale adaptive Siamese network based trackers on the scale variation (SV) attribute of the online tracking benchmark datasets. The highest and the second highest values are highlighted in bold and underlined, respectively.

		OTB2013		**OTB50**		**OTB2015**	
	Method	**Prec.**	**AUC**	**Prec.**	**AUC**	**Prec.**	**AUC**
	SiamFC	0.817	0.615	0.682	0.519	0.735	0.559
SV	C-RPN	<u>0.901</u>	<u>0.667</u>	<u>0.824</u>	<u>0.611</u>	<u>0.859</u>	<u>0.647</u>
	CAT	**0.926**	**0.708**	**0.884**	**0.665**	**0.899**	**0.687**

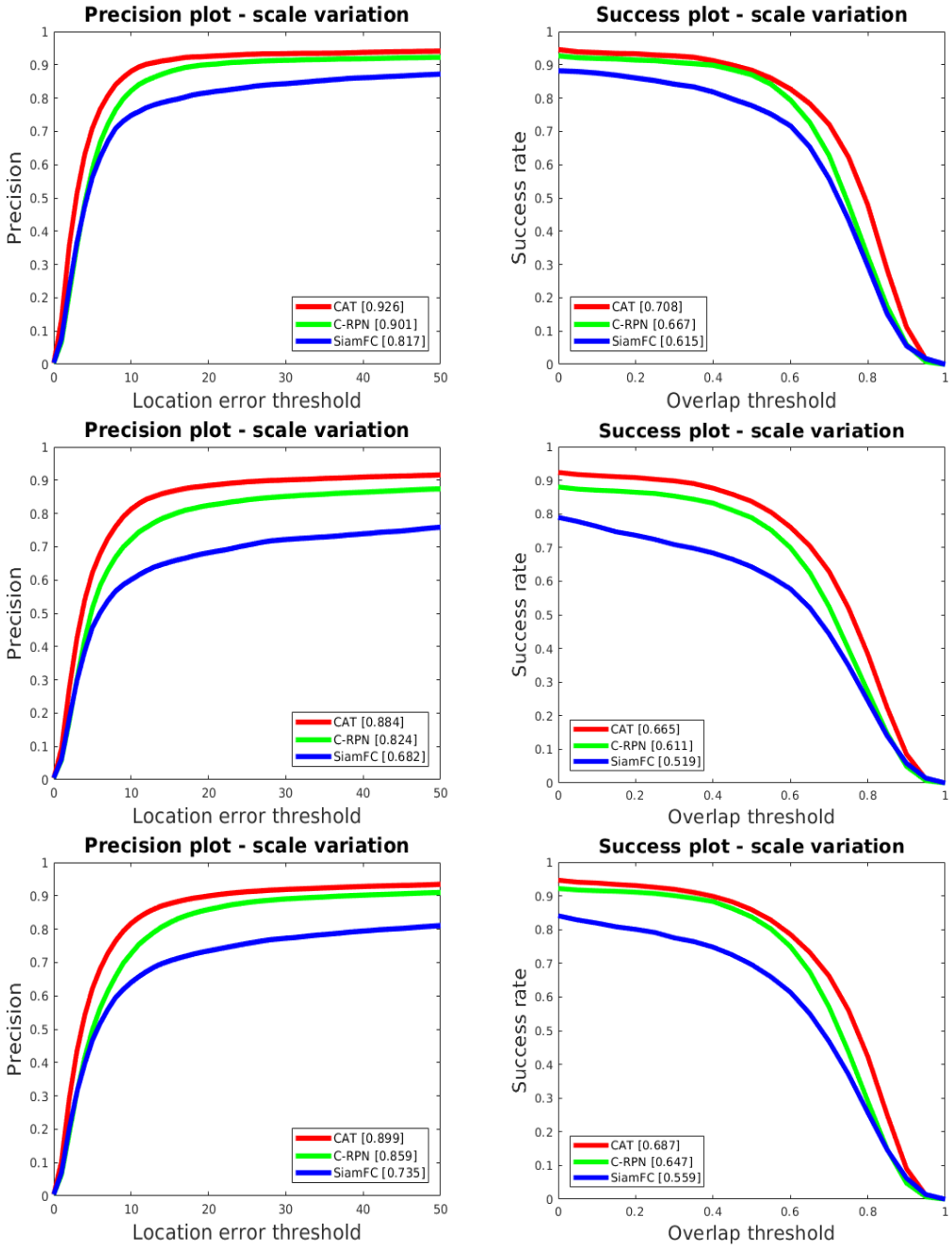

Figure 5.4: The precision plots and success plots of CAT and two representative scale adaptive Siamese network based trackers on the scale variation attribute of the online tracking benchmark datasets. From top to bottom, the plots correspond to OTB2013, OTB50 and OTB2015, respectively. The left column corresponds to the precision plots and the right column corresponds to the success plots.

To verify the effectiveness of our proposed tracker when dealing with aspect ratio variation challenges, we perform experiments on the aspect ratio variation attributes of the OTB datasets. The performance is summarized in Table 5.3. The precision plots and success rate plots for the scale variation attribute are presented in Figure 5.5.

As shown in Table 5.3 and Figure 5.5, our proposed tracker CAT obtains the best performance on all OTB datasets. In comparison with C-RPN, which obtains the second best performance, CAT improves the precision score by 4.0%, 7.7%, and 6.8% and the AUC score by 3.1%, 5.3%, 3.9% on OTB2013, OTB50, and OTB2015, respectively. The significant performance improvement demonstrates that our proposed tracker CAT can accommodate aspect ratio variations effectively.

Table 5.3: Comparisons between CAT and two representative scale adaptive Siamese network based trackers on the aspect ratio variation attribute (ARV) of online tracking benchmark datasets. The highest and the second highest values are highlighted in bold and underlined, respectively.

		OTB2013		**OTB50**		**OTB2015**	
	Method	**Prec.**	**AUC**	**Prec.**	**AUC**	**Prec.**	**AUC**
ARV	SiamFC	0.745	0.534	0.605	0.454	0.673	0.491
	C-RPN	<u>0.832</u>	<u>0.592</u>	<u>0.757</u>	<u>0.551</u>	<u>0.791</u>	<u>0.584</u>
	CAT	**0.872**	**0.623**	**0.834**	**0.604**	**0.859**	**0.623**

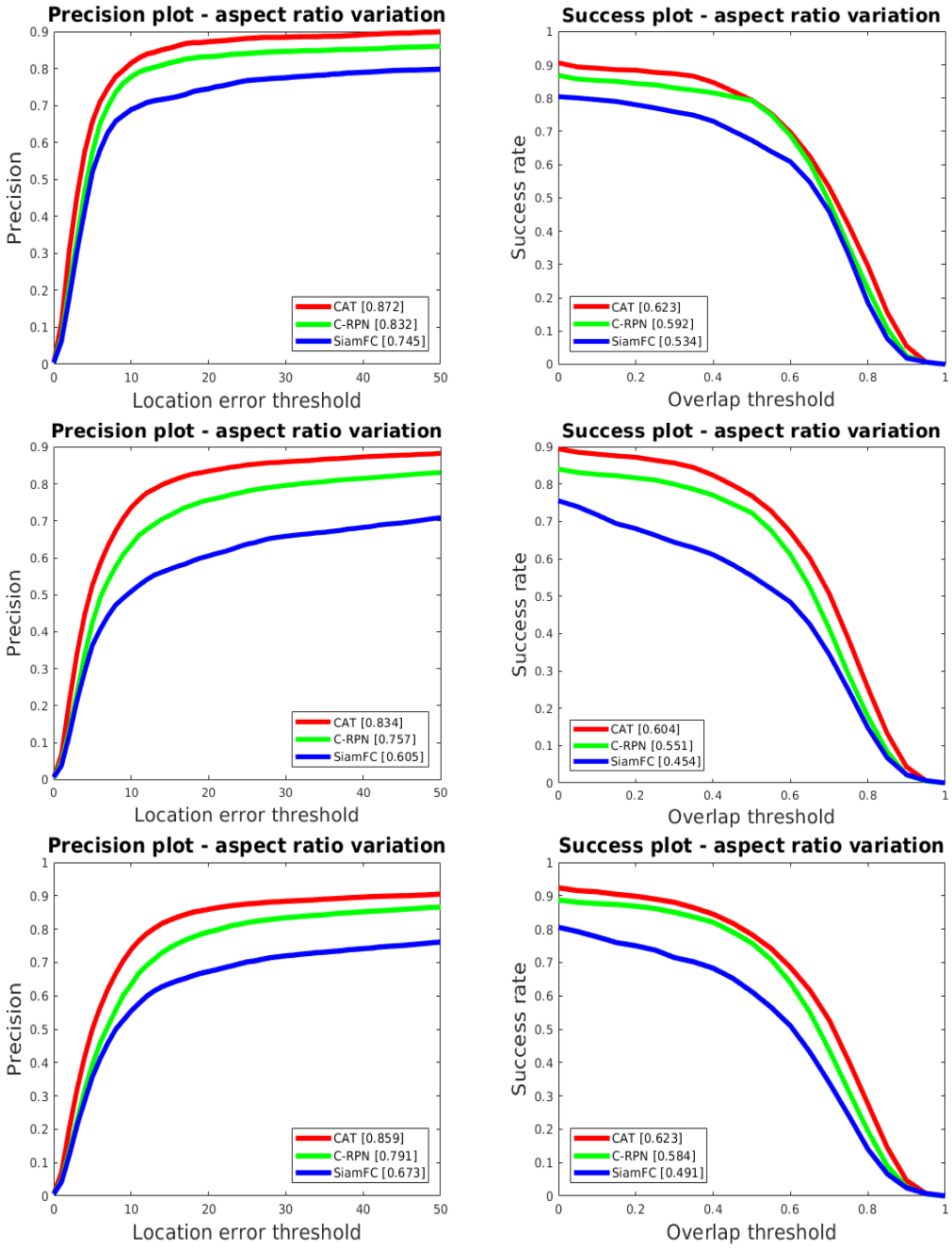

Figure 5.5: The precision plots and success plots of CAT and two representative scale adaptive Siamese network based trackers on the aspect ratio variation attribute of the online tracking benchmark datasets. From top to bottom, the plots correspond to OTB2013, OTB50 and OTB2015, respectively. The left column corresponds to the precision plots and the right column corresponds to the success plots.

5.4.4 Internal Analysis of the Proposed Method

In order to reveal the contributions of each component of CAT, we design three variants of CAT and perform a component-wise analysis on OTB2015. In these variants, the regression branches for state estimation are kept as the same and trained using the intersection over union loss, and other branches are trained with standard cross-entropy loss if not specifically mentioned. The three variants of our proposed tracker are described as follows and the results are summed up in Table 5.4.

- CAT_wo_cen: The CAT tracker without the centerness-aware classification branch is denoted as CAT_wo_cen, where a classification branch trained for binary foreground-background identification is employed instead of our proposed centerness-aware classification branch.
- CAT_w_cen_div: The CAT tracker with a separate branch trained for centerness value estimation is denoted as CAT_w_cen_div, in which a single-layer branch paralleling the binary classification branch is trained to estimate the centerness value.
- CAT_wo_mod: The CAT tracker with the proposed centerness-aware classification branch trained by the standard cross entropy loss without the modulating factor.

Table 5.4: Comparisons between CAT and its three variants on OTB2015. The highest and the second highest scores are in boldface and underlined, respectively.

	CAT_wo_cen	CAT_w_cen_div	CAT_wo_mod	CAT
Prec.	0.615	0.836	0.837	**0.882**
AUC	0.486	0.640	0.645	**0.670**

As presented in Table 5.4, the centerness branch yields a significant gain, *i.e.*, 15.4 points on the AUC score and 22.1 points on precision score, which verifies the effectiveness of the centerness. By

suppressing the low quality predictions, the tracker avoids the low quality predictions that may result in tracking failures due to the error accumulation. As shown in the comparison between CAT and CAT_wo_mod, facing the challenges in the training process as described in Section 5.3.4, the loss function that combines the standard cross-entropy loss and the modulating factor ensures the successful training of our tracker with the proposed centerness-aware classification branch, and yields a significant gain. As presented in the comparison between CAT and CAT_w_cen_div, the proposed centerness-aware classification branch brings about a promising performance improvement of 3.0 points on AUC score and 4.6 points on the precision score, which demonstrates that the proposed centerness-aware classification branch mitigates the inconsistency between the training and the inference stages by acting as a joint representation of the binary classification branch and the centerness branch. These comparisons verify the efficacy of different components of CAT.

5.5 Discussion

In this chapter, we design an effective centerness-aware anchor-free tracking framework that avoids the heuristic design associated with anchor boxes and fixed set of scale factors. Based upon the observation that location near the center of the target can provide high quality predictions, we propose a centerness-aware classification branch to suppress the low quality state estimations. Extensive experiments show that the proposed tracker CAT operates at frame-rates beyond real-time and obtains salient performance in a wide variety of tracking scenarios.

6

Conclusions and Future Work

6.1 Conclusions

In Chapter 2 to Chapter 5, this dissertation presents four visual trackers. Based on the two most prominent paradigms for visual object tracking, *i.e.*, discriminative correlation filter based trackers and Siamese network based trackers, scale adaptive trackers are proposed to address the scale variation challenges under different tracking scenarios, and can be employed to track the target object or

incorporated into other trackers to further boost tracking performance in terms of solution quality and computational expense. Specifically, based on the discriminative correlation filter, by exploiting the robustness of a novel criteria and investigating different computational cost reduction strategies, SITUP and FAST can meet the demands of scale variation challenges effectively while maintaining real-time running speed on a single CPU. Further, TARA enables aspect ratio adaptability via using a group of discriminative correlation filters to localize the boundary of the target object. In comparison with SITUP and FAST, TARA provides promising performance improvement but cannot be applied in real-time applications due to the employed deep hierarchical convolutional features. Based on the Siamese network, CAT obviates the need for the predefined candidate boxes and the associated parameters, and obtains salient tracking performance by exploiting the expressive power of the fully convolutional network. Extensive empirical evaluation on the publicly available tracking benchmark datasets demonstrates that the proposed trackers can meet the demands of scale variation challenges effectively while maintaining promising tracking performance.

6.2 Future Work

Besides the two main paradigms mentioned in this dissertation, there is a trend to make the tracker learn through a trial-and-error approach, and the main paradigm implementing such a mechanism is reinforcement learning. Recently, deep reinforcement learning based tracking[78–84] has grown rapidly as an area of active research. One pioneering work is proposed in reference 78, in which the state is denoted as the cropped target image based on the bounding box detected at the previous frame. The actor network is trained to provide continuous actions to update the previous bounding box to provide a tight bounding box for the target object in the current frame. However, the reward function is designed to provide the reward value as 1 if the intersection-over-union value between the estimated bounding box and ground truth bounding box is higher than a predefined thresh-

old value. The simple reward function ignores other geometric factors, hence hinders the optimal tracking performance. The improvement will be based on how to design the reward function for reinforcement learning based trackers by taking both the distance penalty and contextual information into consideration.

Another future direction of this thesis involves enabling template update in CAT. As illustrated in Section 5.3.1, the Siamese network takes an exemplar image and a search image as input, and the image patch with the center point as the target center is cropped in the initial frame, and is then adopted as the exemplar image. The template patch is kept fixed and not updated since an inaccurate tracking result may pollute the template and yield inferior tracking performance or even result in losing the target object. However, a fixed exemplar image cannot provide accurate tracking result if the target object experiences extreme appearance variations. The improvement will be based on how to identify the confidence level of the tracking result, and then update the exemplar image to better adapt to the appearance variation of the target object accordingly.

6.3 Publications

- **Haoyi Ma**, Scott T. Acton, and Zongli Lin, "OSLO: Automatic Cell Counting and Segmentation for Oligodendrocyte Progenitor Cells," in *25th IEEE International Conference on Image Processing*, 2018, pp. 2431-2435.

- **Haoyi Ma**, Scott T. Acton, and Zongli Lin, "SITUP: Scale Invariant Tracking using Average Peak-to-correlation Energy," *IEEE Transactions on Image Processing*, vol. 29, no. 1, pp. 3546-3557, 2020.

- **Haoyi Ma**, Zongli Lin, and Scott T. Acton, "FAST: Fast and Accurate Scale Estimation for Tracking," *IEEE Signal Processing Letters*, vol. 27, no. 1, pp. 161-165, 2019.

- **Haoyi Ma**, Scott T. Acton, and Zongli Lin, "TARA: Tracking with Aspect Ratio Adaptability," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 847-851.

- **Haoyi Ma**, Scott T. Acton, and Zongli Lin, "CAT: Centerness-Aware Anchor-Free Tracker," *Sensors*, vol. 22, no. 1, p. 354, 2022.

References

[1] Arun Hampapur, Lisa Brown, Jonathan Connell, Ahmet Ekin, Norman Haas, Max Lu, Hans Merkl, and Sharath Pankanti. Smart video surveillance: exploring the concept of multi-scale spatiotemporal tracking. *IEEE signal processing magazine*, 22(2):38–51, 2005.

[2] Laurent Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE transactions on image processing*, 13(10):1304–1318, 2004.

[3] Nilanjan Ray, Scott T Acton, and Klaus Ley. Tracking leukocytes in vivo with shape and size constrained active contours. *IEEE transactions on medical imaging*, 21(10):1222–1235, 2002.

[4] Jing Cui, Scott T Acton, and Zongli Lin. A monte carlo approach to rolling leukocyte tracking in vivo. *Medical Image Analysis*, 10(4):598–610, 2006.

[5] Xinmin Liu, Zongli Lin, and Scott T Acton. A grid-based bayesian approach to robust visual tracking. *Digital Signal Processing*, 22(1):54–65, 2012.

[6] Dipti Prasad Mukherjee, Nilanjan Ray, and Scott T Acton. Level set analysis for leukocyte detection and tracking. *IEEE Transactions on Image processing*, 13(4):562–572, 2004.

[7] Haoyi Ma, Rebecca Beiter, Alban Gaultier, Scott T Acton, and Zongli Lin. Oslo: Automatic cell counting and segmentation for oligodendrocyte progenitor cells. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2431–2435. IEEE, 2018.

[8] Hao Li, Tianhao Xiezhang, Cheng Yang, Lianbing Deng, and Peng Yi. Secure video surveillance framework in smart city. *Sensors*, 21(13):4419, 2021.

[9] Rolf H Baxter, Michael JV Leach, Sankha S Mukherjee, and Neil M Robertson. An adaptive motion model for person tracking with instantaneous head-pose features. *IEEE Signal Processing Letters*, 22(5):578–582, 2014.

[10] 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):125–141, 2008.

[11] Xue Mei and Haibin Ling. Robust visual tracking using ℓ 1 minimization. In *2009 IEEE 12th international conference on computer vision*, pages 1436–1443. IEEE, 2009.

[12] Allan D Jepson, David J Fleet, and Thomas F El-Maraghi. Robust online appearance models for visual tracking. *IEEE transactions on pattern analysis and machine intelligence*, 25(10): 1296–1311, 2003.

[13] Shaohua Kevin Zhou, Rama Chellappa, and Baback Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Transactions on Image Processing*, 13(11):1491–1506, 2004.

[14] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L Hicks, and Philip HS Torr. Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2096–2109, 2015.

[15] Jianming Zhang, Shugao Ma, and Stan Sclaroff. Meem: robust tracking via multiple experts using entropy minimization. In *European conference on computer vision*, pages 188–203. Springer, 2014.

[16] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2544–2550. IEEE, 2010.

[17] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[18] Horst Possegger, Thomas Mauthner, and Horst Bischof. In defense of color-based model-free tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2113–2120, 2015.

[19] 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(3):583–596, 2014.

[20] Joao F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *European conference on computer vision*, pages 702–715. Springer, 2012.

[21] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016.

[22] Vladimír Ulman, Martin Maška, Klas EG Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miroslav Radojevic, et al. An objective comparison of cell-tracking algorithms. *Nature methods*, 14(12):1141–1152, 2017.

[23] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost Van de Weijer. Adaptive color attributes for real-time visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1090–1097, 2014.

[24] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. Bmva Press, 2014.

[25] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *European conference on computer vision*, pages 254–265. Springer, 2014.

[26] Mengdan Zhang, Junliang Xing, Jin Gao, and Weiming Hu. Robust visual tracking using joint scale-spatial correlation filters. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1468–1472. IEEE, 2015.

[27] Andres Solis Montero, Jochen Lang, and Robert Laganiere. Scalable kernel correlation filter with sparse feature integration. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 587–594. IEEE, 2015.

[28] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013.

[29] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015. doi: 10.1109/TPAMI.2014.2388226.

[30] Robert M Gray. Toeplitz and circulant matrices: A review. 2006.

[31] Fahad Shahbaz Khan, Rao Muhammad Anwer, Joost Van De Weijer, Andrew D Bagdanov, Antonio M Lopez, and Michael Felsberg. Coloring action recognition in still images. *International journal of computer vision*, 105(3):205–221, 2013.

[32] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr. Staple: Complementary learners for real-time tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1401–1409, 2016.

[33] Stephen Siena and BVK Vijaya Kumar. Detecting occlusion from color information to improve visual tracking. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1110–1114. IEEE, 2016.

[34] Mengmeng Wang, Yong Liu, and Zeyi Huang. Large margin object tracking with circulant feature maps. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4021–4029, 2017.

[35] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692, 2015.

[36] Haoyi Ma, Scott T Acton, and Zongli Lin. Situp: Scale invariant tracking using average peak-to-correlation energy. *IEEE Transactions on Image Processing*, 29:3546–3557, 2020.

[37] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[38] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[40] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[42] Haoyi Ma, Zongli Lin, and Scott T Acton. Fast: Fast and accurate scale estimation for tracking. *IEEE Signal Processing Letters*, 27:161–165, 2019.

[43] Dafei Huang12, Lei Luo, Mei Wen12, Zhaoyun Chen12, and Chunyuan Zhang12. Enable scale and aspect ratio adaptability in visual tracking with detection proposals. 2015.

[44] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.

[45] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, Jongwoo Lim, and Ming-Hsuan Yang. Hedged deep tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4303–4311, 2016.

[46] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 3074–3082, 2015.

[47] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. Robust fragments-based tracking using the integral histogram. In *2006 IEEE Computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 1, pages 798–805. IEEE, 2006.

[48] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[50] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Robust visual tracking via hierarchical convolutional features. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2709–2723, 2018.

[51] Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7952–7961, 2019.

[52] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[53] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8971–8980, 2018.

[54] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 101–117, 2018.

[55] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[56] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 1763–1771, 2017.

[57] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4854–4863, 2018.

[58] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. A twofold siamese network for real-time object tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4834–4843, 2018.

[59] Guangting Wang, Chong Luo, Zhiwei Xiong, and Wenjun Zeng. Spm-tracker: Series-parallel matching for real-time visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3643–3652, 2019.

[60] Ze Yang, Yinghao Xu, Han Xue, Zheng Zhang, Raquel Urtasun, Liwei Wang, Stephen Lin, and Han Hu. Dense reppoints: Representing visual objects with dense point sets. In *European Conference on Computer Vision*, pages 227–244. Springer, 2020.

[61] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[62] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 850–859, 2019.

[63] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 850–859, 2019.

[64] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[65] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 840–849, 2019.

[66] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, Lei Li, and Jianbo Shi. Foveabox: Beyound anchor-based object detection. *IEEE Transactions on Image Processing*, 29:7389–7398, 2020.

[67] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[68] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8514–8523, 2021.

[69] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6269–6277, 2020.

[70] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[71] Haoyi Ma, Scott T Acton, and Zongli Lin. Tara: Tracking with aspect ratio adaptability. In *2020 54th Asilomar Conference on Signals, Systems, and Computers*, pages 847–851. IEEE, 2020.

[72] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520, 2016.

[73] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

[74] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[75] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5296–5305, 2017.

[76] Jia Deng, Wei Dong, Richard Socher, Li-Jia 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. Ieee, 2009.

[77] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1562–1577, 2019.

[78] Boyu Chen, Dong Wang, Peixia Li, Shuang Wang, and Huchuan Lu. Real-time'actor-critic'tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 318–334, 2018.

[79] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2711–2720, 2017.

[80] Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking and its real-world deployment via reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1317–1332, 2019.

[81] Zhao Zhong, Zichen Yang, Weitao Feng, Wei Wu, Yangyang Hu, and Cheng-Lin Liu. Decision controller for object tracking with deep reinforcement learning. *IEEE Access*, 7:28069–28079, 2019.

[82] Yifan Jiang, Hyunhak Shin, and Hanseok Ko. Precise regression for bounding box correction for improved tracking based on deep reinforcement learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1643–1647. IEEE, 2018.

[83] Tian Wang, Ruoxi Qin, Yang Chen, Hichem Snoussi, and Chang Choi. A reinforcement learning approach for uav target searching and tracking. *Multimedia Tools and Applications*, 78(4):4347–4364, 2019.

[84] Bineng Zhong, Bing Bai, Jun Li, Yulun Zhang, and Yun Fu. Hierarchical tracking by reinforcement learning-based searching and coarse-to-fine verifying. *IEEE Transactions on Image Processing*, 28(5):2331–2341, 2018.