Visual Object Tracking

Zijing Chen

Faculty of Engineering and Information Technology

University of Technology Sydney

A thesis submitted for the degree of

Doctor of Philosophy

2018

Certificate of Original Authorship

I, Zijing Chen, declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This thesis is the result of a research candidature conducted with another University as part of a collaborative Doctoral degree.

This document has not been submitted for qualifications at any other academic institution.

Production Note:

Signature: Signature removed prior to publication.

Date: 29/10/2018

Acknowledgements

I would like to express my gratitude to all those who helped me finish my doctoral studies.

First and foremost, I want to extend my heartfelt gratitude to my supervisor, Professor Dacheng Tao. His kind supervision, valuable suggestions and warm encouragement helped me to overcome difficulties and to successfully complete this thesis. His foresight and sagacity inspired me to find inspiration in cutting-edge research, which made the process of research a great joy. He deserves to be admired, not only for his great achievements in artificial intelligence, computer vision and machine learning, but for his diligence and courage which inspires others to expend a greater effort on their research.

Second, I would like to thank my co-supervisor, Dr Jun Li, who helped me pursue my passion for research. With incredible talent, he is full of inspiring and effective ideas which helped me survive the tedious and burdensome research tasks. He is extremely patient and is generous with his time, helping me to make gradual progress.

Third, I wish to express my sincere appreciation to Prof. Xinge You. He is a Dual-PhD supervisor at Huazhong University of Science and Technology. Although we are far away from each other, Prof. You gave me many suggestions and ideas via emails and numerous phone calls.

Fourth, I would like to pay tribute to Prof. Sean He, Dr Ling Chen, Dr Wei Liu, and Dr Guoqiang Zhang for their inspiring words, continued support, and elaborate instructions about my presentations and academic reports. They had a significantly positive influence on me during my Ph.D. career.

I would like to thank my excellent collaborators. Dr Ruxin Wang and Dr Qiang Li helped me both in my research work and life in Sydney. Dr Xiubao Jiang, Dr Weihua Ou, Dr Hasan Abdulwahid gave me many tips to help me find solutions, especially in solving complicated mathematical problems related to my research. Mr Huan Fu and Mr Baosheng Yu guided me to the right path on deep learning. I also worked closely with Mr Boxuan Zhong who is always optimistic and cheerful.

Special thanks to my husband Zhe Chen, who helped me to overcome depression, insomnia and pessimism. He was the one who encouraged me to grow as an individual and helped me to replace insecurity with confidence.

Finally yet importantly, I would like to express my special thanks to my parents. I really appreciate their support and the confidence they have placed in me.

Abstract

Visual object tracking is a critical task in many computer-vision-related applications, such as surveillance and robotics. If the tracking target is provided in the first frame of a video, the tracker will predict the location and the shape of the target in the following frames. Despite the significant research effort that has been dedicated to this area for several years, this field remains challenging due to a number of issues, such as occlusion, shape variation and drifting, all of which adversely affect the performance of a tracking algorithm.

This research focuses on incorporating the spatial and temporal context to tackle the challenging issues related to developing robust trackers. The spatial context is what surrounds a given object and the temporal context is what has been observed in the recent past at the same location. In particular, by considering the relationship between the target and its surroundings, the spatial context information helps the tracker to better distinguish the target from the background, especially when it suffers from scale change, shape variation, occlusion, and background clutter. Meanwhile, the temporal contextual cues are beneficial for building a stable appearance representation for the target, which enables the tracker to be robust against occlusion and drifting.

In this regard, we attempt to develop effective methods that take advantage of the spatial and temporal context to improve the tracking algorithms. Our proposed methods can benefit three kinds of mainstream tracking frameworks, namely the template-based generative tracking framework, the pixel-wise tracking framework and the tracking-by-detection framework. For the template-based generative

tracking framework, a novel template based tracker is proposed that enhances the existing appearance model of the target by introducing mask templates. In particular, mask templates store the temporal context represented by the frame difference in various time scales, and other templates encode the spatial context. Then, using pixel-wise analytic tools which provide richer details, which naturally accommodates tracking tasks, a finer and more accurate tracker is proposed. It makes use of two convolutional neural networks to capture both the spatial and temporal context. Lastly, for a visual tracker with a tracking-by-detection strategy, we propose an effective and efficient module that can improve the quality of the candidate windows sampled to identify the target. By utilizing the context around the object, our proposed module is able to refine the location and dimension of each candidate window, thus helping the tracker better focus on the target object.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Since the costs of high-quality cameras have dropped dramatically, video surveillance systems are nowadays widely used in public spaces. In order to handle the increasing amount of captured video data, it is essential to utilize automated video processing techniques [41]. Of the various popular video processing techniques, the visual object tracking technique plays a critical role in a wide range of applications such as surveillance and robotics.

The goal of visual tracking is to identify the states of a moving object in all the frames of a video sequence given only the initial state of the target. Commonly, the state of an object in each frame can be presented in two formats: 1) a bounding box that marks a rectangular region, and 2) a binary mask that highlights an area at the pixel level. In practice, a bounding box generally uses four numbers to describe the size and shape of an object. Due to efficiency, the bounding box is a widely used format to denote the region of interest (ROI) in many computer vision tasks such as object detection [133] and action recognition [155]. In addition to the bounding box, the binary mask is another popular format to denote the ROI. Instead of only denoting a rectangular area, a binary mask provides a pixel-wise segmentation of the foreground area to identify objects. Although such pixel-level segmentation of the objects could be more time-consuming during tracking [116], it can measure the target states in detailed contours and shapes,

thus achieving a more precise estimation of the target in each frame.

Figure 1.1 shows an example of object states in visual object tracking. In the figure, the first row presents the frames of a video sequence starting from time $t = 0$, and the second row lists the ground truth states of the target in the corresponding frames. The red rectangle in the bottom-left picture represents the initial state of the target, i.e., a car in this example, in the form of a bounding box, and the yellow rectangles in other pictures denote the desired output of a tracking algorithm. Meanwhile, the states of the target in the form of a binary mask are illustrated by the white areas in each frame in the second row. Using the binary mask format, a tracker is supposed to assign 1 to every pixel on the target and 0 to all other areas.

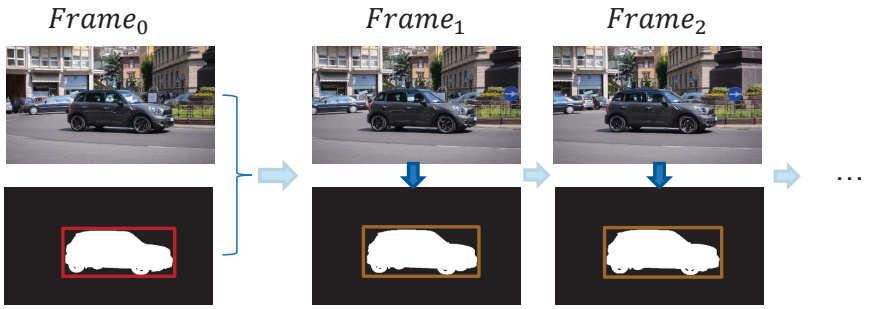


Figure 1.1: An example of object states in online object tracking. The first row presents a few frames starting from time $t = 0$, and the second row lists the ground truth states of the target correspondingly. The states in the form of a bounding box are illustrated with rectangles, while the states in the form of binary masks are illustrated in white areas. Either the red rectangle or the white mask inside represents the initial state of the target. Correspondingly, the yellow rectangles or the related masks are the desired output of a tracker in each new frame.

Over the past few decades, various methods have been proposed to tackle the tracking problem, and promising results have been achieved on different benchmarks. In general, researchers mainly study two groups of tracking algorithms, which are generative methods and discriminative methods. Generative methods attempt to build a robust appearance model of the target. In each new frame, candidate windows are sampled around a previously estimated location of the target. Then, according to a maintained appearance model, the window whose

image content is the most similar to the target will be considered as the new state of the target. Based on this methodology, researchers have introduced diversified algorithms to tackle the tracking problem. Some of the most popular generative algorithms include [88, 112]. Different from generative methods, given a set of candidate windows, discriminative methods then try to exploit the discriminative ability of a classifier to distinguish the foreground window from background windows. In particular, these methods mainly train a target-specific classifier on-the-fly. For the sampled candidate windows in a new frame, the trained classifier will provide the foreground/background score for the windows. Then the window with the highest score could be chosen as the new state of the target. The most representative methods of this kind are correlation filter-based trackers and convolutional neural network-based trackers. The correlation filter-based trackers, such as [70, 96], train a correlation filter as the target-specific classifier. By exploiting the properties of circular correlation and performing the correlation operations in the Fourier domain, the correlation filter-based trackers are extremely efficient and can achieve compelling tracking accuracy. In addition to the correlation filter, researchers also attempt to take advantage of the impressive expression capacity of Deep Convolutional Neural Networks (DCNNs) to tackle the tracking problem. In recent years, DCNNs have demonstrated their outstanding performance in almost all mainstream computer vision tasks, especially classification. DCNNs have achieved outstanding performance on various challenges and sometimes they even surpass human annotators. Hence, many researchers tend to incorporate DCNNs in their tracking models. In particular, MDNet, which trains a DCNN as the target-specific classifier, achieved first place in the visual object tracking challenge (VOT15 [110]). Afterward, more powerful algorithms were introduced in the field of tracking with more intricate network structures or additional target cues, for example, methods based on Siamese networks [10, 153], and algorithms with complementary deep information like motion cues [199].

Despite the great research effort over the last several years, several issues remain challenging. For example, scale change, shape variation, occlusion, and background clutter adversely affect the performance of a tracking algorithm significantly [146] [179]. Figure 1.2 presents two video sequences from [128] that

contain some of these challenges. The first row of the figure shows a video clip that records a soapbox race. The target, which is the vehicle and two drivers as a whole, suffers from scale change and shape variation because of the translation of the distance and pose of the target with respect to the camera. The sequence in the second row indicates a dog that is running in a garden with clutter background. We can see that barriers in some frames occlude the dog, making it difficult for trackers to distinguish the target from the background. Other challenging issues presented in this example include illumination change and cluttered background surroundings. In addition to these issues, there are many other challenging issues such as non-rigid deformation, motion blur, in-plane rotation, out-of-plane rotation, lost and re-appear, and so on. Due to these issues, a tracker would easily drift to other irrelevant objects and thus fail to track the target successfully.

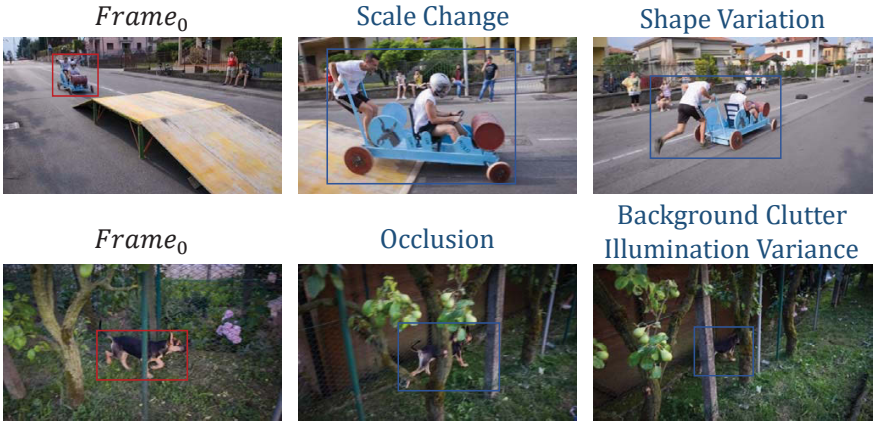


Figure 1.2: Example of challenging issues in object tracking. The target in the first row suffers from scale change and shape variation. The target in the second row suffers heavy occlusions, illumination changes and cluttered background surroundings.

In order to tackle these issues, spatial and temporal visual contexts could be advantageous for visual tracking. In particular, the visual contexts that surround the target or in the preceding time periods could contain complementary information to resolve the confusion caused by issues such as scale change, shape variation, and occlusion. For example, although the dog in Fig. 1.2 is partially occluded, a tracker would still be able to infer the holistic appearance of the

target by referring to the surrounding areas.

Some studies have incorporated contexts to help tackle tracking. Researches like [70, 71] simply crop a larger candidate window to include more spatial contexts. The Context tracker [42] explores the visual context in recent frames and nearby areas to build an accurate appearance model. By analysing the context information, it avoids the distraction from nearby objects which may have a similar appearance with the tracking target. The studies in [72,107] develope a long-term module which stores target information in past frames as temporal contexts to facilitate the tracking for the current frame. However, it is still challenging for these studies to deliver robust tracking. To this end, we aim at proposing algorithms which can effectively and extensively incorporate contexts in different tracking scenarios. In the next section, we introduce how we build a robust tracker by referring to informative contextual information.

1.2 Motivation of This Study

Both neurophysiological [141] and statistical analysis on typical natural scenes and movies [48, 194] demonstrate that the visual processing of objects is powerfully affected by its context, that is, its spatial and temporal neighbourhood. By studying the mechanism of visual processing of human brains, the work published in Neuroscience [141] finds that the human brain uses the complete spatio-temporal input $\mathbf{I}$ to process the visual world. In particular, the work in [50] studies the local association field of the human visual system and discovers that most objects in the visual world have large spatial and temporal footprints. This implies that there exists correlations between $\mathbf{I}_{s1}(t_1)$ and $\mathbf{I}_{s2}(t_2)$ where $s1$ and $s2$ represent image areas that are spatially near to each other and t_1 and t_2 represent different time steps that are close. In addition, the work in [48], in which the context information is analysed in the way shown in Figure 1.3, collects and analyses image statistics from the real-life Catcam movie database. For spatial statistics, the patches (denoted by orange circles) give the context for surrounding spatial positions. For temporal statistics, context patches of the neighbouring time frames of the sequence in the same spatial location are collected. This research discovers that the statistics of videos for space and time reveal how neighbouring

spatial and temporal locations are correlated. Existing theoretical and empirical studies indicate that it is highly beneficial to take contextual information into account in visual tracking.

The visual context can help a tracker to better locate the target since the visual appearance of a moving target strongly depends on both spatial context and temporal context. More specifically, the spatial context provides information about what surrounds a given target, making it easier to distinguish the target from its surroundings. The temporal context then provides information about what has been observed in the recent past and can properly reveal the changes in the appearance of the target over time. As a result, combining the context information is beneficial for building a more accurate and more robust appearance model against various challenging issues, including scale change, shape variation, occlusion, and background clutter. Additional contextual information also contributes extra target details to facilitate the tracker. Finally, contextual information can reduce the risk of accumulating tracking errors because it enables the tracker to identify and rectify inappropriate tracking results in a more timely manner. Hence, it is less likely that the tracker will drift away from the target. Therefore, this thesis aims at developing robust tracking models by effectively incorporating rich spatial-temporal context.

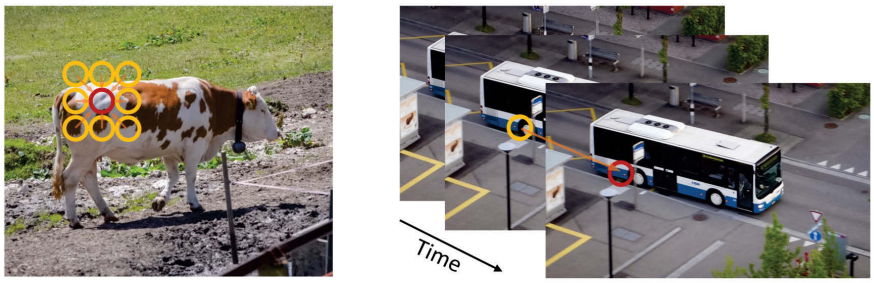


Figure 1.3: Spatial and temporal context. The patches (denoted by orange circles) give spatial context for surrounding spatial positions. For temporal context, the context patches of the neighbouring time frames of the sequence in the same spatial location are collected.

1.3 Summary of Contributions

This thesis focuses on category-independent single object online tracking, in which we build single generic tracker for all kinds of objects. Given only the initial position, the on-line tracking methods aim to track objects in the subsequent frames of a video. There is no preliminary information about the target object. By incorporating the spatial and temporal context, robust trackers can be developed, which are powerful in tackling scale change, shape variation, occlusion, and background clutter.

First, we design a template-based algorithm for tracking the target, which is more robust against occlusion. The proposed method enhances the existing sparse appearance model by introducing mask templates produced by frame difference. Then, object changes in different time scales can be successfully represented by the mask templates, especially when the target is occluded. There are two innovations: 1) We introduce a set of templates which not only encodes the context into the sparse appearance model but also dramatically reduces the dimension of the ℓ_1 minimisation problem. Furthermore, when we model the temporal evolution of the system, we boost the performance of our tracker by considering the system dynamic, namely state estimation. 2) We demonstrate that the adapted problem with template regulation can be solved efficiently using the Accelerated Proximal Gradient (APG) algorithm. In doing so, we both increase tracking accuracy and relieve the computational burden.

Second, we build a novel bi-channel fully convolutional network for accurate pixel-level visual object tracking by utilising the temporal and spatial context. With the bi-channel structure, where the low-level optical flow branch captures temporal context while the high-level branch represents semantic change, the network naturally encodes the spatial-temporal context of the target. This network outputs pixel-level tracking results which are finer and more accurate and is more robust against shape variation and background clutter than traditional bounding-box-based methods.

Third, under a tracking-by-detection framework, we propose a module that can refine the candidate windows for trackers and improve the tracking performance. By analysing the spatial context around each candidate window, our

proposed module rectifies the location and shape of the window. Thus, the refined window will be able to cover the target more precisely and tightly. With this module, the tracker will be able to better focus on the target and output a more accurate tracking result in each frame. In addition, the utilization of the proposed module is flexible. It can be easily embedded into the existing tracking-by-detection framework to boost the performance of the tracker, no matter whether the tracker is based on traditional features or developed with CNN features, and the time overhead is negligible.

This thesis is organized as follows. Chapter 2 provides a literature review of the works most related to our tracking algorithms. Chapter 3, Chapter 4 and Chapter 5 introduce three proposed trackers that correspond to the aforementioned contributions respectively. Finally, Chapter 6 presents the conclusions and future work.

1.4 Publications Related to the Thesis

My publications related to this thesis are listed below:

1. **Zijing Chen**, Xinge You, Boxuan Zhong, Jun Li, Dacheng Tao. Dynamically Modulated Mask Sparse Tracking. IEEE Trans. Cybernetics 47(11): 3706-3718 (2017)

2. **Zijing Chen**, Jun Li and Xinhua You. Learn to Focus on Objects for Visual Detection. Accepted by NeuroComputing.

3. **Zijing Chen**, Jun Li, Zhe Chen, Xinge You. Generic Pixel Level Object Tracker Using Bi-Channel Fully Convolutional Network. ICONIP (1) 2017: 666-676

4. **Zijing Chen**, Xinhua You and Jun Li. Learning to focus for object proposals. SPAC 2017 439-444.

5. Jun Li, **Zijing Chen**, Zhenyuan Ma. Learning Colours from Textures by Sparse Manifold Embedding. Advances in Signal Processing: Reviews, Book Series, Vol. 1.

6. Zhe Chen, **Zijing Chen**. RBNet: A Deep Neural Network for Unified Road and Road Boundary Detection. ICONIP (1) 2017: 677-687

7. Boxuan Zhong, **Zijing Chen**, Xinge You, Luoqing Li, Yunliang Xie, Shujian Yu. Robust weighted coarse-to-fine sparse tracking. SPAC 2014: 7-14

Chapter 2

Literature Review

This chapter provides a comprehensive overview of existing tracking methods. Meanwhile, we also include a brief review of research areas related to the content of the proposed methods, like object detection and video segmentation.

There is a number of good review works [21, 95, 146]. We mainly cover studies that are closely related to my research in tracking. We follow the survey [95, 146] and phrase typical visual object tracking system in four stages: object initialization, appearance modeling, motion estimation, and object localization. In our work, we focus on the appearance modeling stage.

The appearance modeling stage is generally composed of visual representation part and statistical modeling part. The visual representations capture global and local visual information. Typical global information representations include vector-based raw pixel representation which used in Incremental Learning for Tracking (IVT) [135], optical flow representation which used in Parallel Robust Online Simple Tracking (PROST) [138], haar feature based representation like Multiple Instance Learning (MIL) [5, 6], Compressive Tracking (CT) [187] and Structured output tracking (Struck) [64]. Typical local information representations include binary pattern based representation like Tracking-Learning-Detection (TLD) [80] [81], ConteXT tracker (CXT) [42], intensity histogram based representation used in Fragment-based tracking (Frag) [1], Color-based Probabilistic tracking (CPF) [129] and so on. In general, they encode global or local statistical characteristics of an image region. The statistical model performs tracking in a tracking-by-detection scheme. It focuses on using different types of

statistical learning schemes to train a robust appearance model for objects.

The performance of a tracker highly depends on the appearance model because a robust appearance model can successfully distinguish the target from the background in complex scenarios [18, 175] such as occlusions [76, 87, 145] and shape variations. As a core component of trackers, the appearance model can be generative, discriminative, or hybrid [121, 196]. Specifically, in generative appearance models, candidates are searched to minimize reconstruction errors. Representative sparse coding is one typical method and has been exploited for visual tracking [79, 100]. In discriminative models [170, 178], tracking is regarded as a classification problem by separating foreground and background.

2.1 Generative Tracking Methods

The generative appearance models mainly concentrate on how to accurately fit the data from the object class [95]. They incrementally learn visual representations for the tracking target region information via online-update mechanisms. In practice, generative trackers often have better descriptive power and demand a small training set. However, they suffer from distractions caused by the background regions with a similar appearance to the object class. The generative appearance models can be divided into Subspace Learning-based models and Kernel-based models [17]. Here we discuss the Subspace Learning-based methods which are most related to our work.

Conventional Subspace Models The subspace learning-based appearance model associates the target with several underlying subspaces. Each of the subspace is spanned by a set of basis templates. In details, suppose τ is the target and $(\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N)$ denotes the templates of an underlying subspace, then the target can be represented by these templates as follows:

$$\tau = (\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N)(c_1, c_2, ..., c_N)^T \tag{2.1}$$

In (2.1), $(c_1, c_2, ..., c_N)$ are the coefficients of basis templates. Thus subspace learning based appearance models focus on how to obtain the basis templates,

as well as the coefficients, of the underlying subspace by using tools for subspace analysis.

Conventional subspace models can use linear or non-linear subspace models. The linear models can use singular value decomposition (SVD) to obtain a closed-form solution to subspace learning. For example, [135] use rank-R singular value decomposition (R-SVD) to build the subspace model with a sample mean update. Also, [167] applies partial least square analysis to learn a low-dimension feature subspace for tracking. For the non-linear models, the nonlinear dimension reduction techniques, such as Local Linear Embedding, is used [98]. In addition, the kernel principal component analysis is constructed to capture the kernelized eigenspace information from the target samples [25].

Sparse Representation The sparse representation model belongs to unconventional subspace models [188]. It represents the target with a linear combination of templates in a dictionary, and has been successfully used in appearance modeling for a target [168, 177, 195]. With this modeling, the observation likelihood for $\mathbf{x}_t$ can be calculated as shown below. In traditional methods based on sparse representation, dictionary templates are divided into two parts: one is called target templates, which describes the rough appearance of target; the other part is called trivial templates, which consists of a set of images with all but one pixel being zero, making a complete and trivial set of bases of the template images. Thus a target candidate $\mathbf{y}$ lies in the linear span of dictionary templates:

$$\mathbf{y} = [\mathbf{T}, \mathbf{I}] \begin{bmatrix} \mathbf{a} \\ \mathbf{e} \end{bmatrix} \triangleq \mathbf{Bc} \tag{2.2}$$

where $\mathbf{B} = [\mathbf{T}, \mathbf{I}]$ is an over-complete dictionary that is made up of target template set $\mathbf{T}$ and trivial template set $\mathbf{I}$. Each column in $\mathbf{T}$ is a target template generated by reshaping pixels of a candidate region into a column vector while each column in $\mathbf{I}$ is a unit vector that has only one nonzero element. The coefficients of templates are generated from objective function which is usually constructed by two functional parts: one helps to get optimal coefficients that can keep the solution close to the measurement; the other promotes sparsity in the solution. One typical example can be found in the $\ell 1$ tracker using Accelerated Proximal

Gradient approach (L1APG) [8], which uses $\ell 1$ term to promote sparsity in the solution and $\ell 2$ term to keep the solution close to the measurements. Since a good target candidate will be represented by templates sparsely, we want to have a sparse solution to Eq. 2.2. This leads to an $\ell 1$ regularized least squares problem, which yields sparse solutions [160].

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \|\mathbf{B}\mathbf{c} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{c}\|_1 \quad , \quad s.t. \quad \mathbf{c} \succeq 0 \tag{2.3}$$

After deriving the solution of $\mathbf{c}$, the observation likelihood of state $\mathbf{x}_t$ is given as

$$p(\mathbf{z}_t|\mathbf{x}_t) = \frac{1}{\Gamma} \exp\{-\alpha||\mathbf{T}\mathbf{a} - \mathbf{y}||_2^2\} \tag{2.4}$$

It reflects the similarity between a target candidate and the appearance of a tracking target represented by the target template set $\mathbf{T}$.

The sparse representation model has two intrinsic limitations that need to be overcome:

(1) The computational complexity

As [8] mentioned, the speed bottleneck is how to solve the $\ell 1$ optimization (Eq. 2.3) much faster, on the scale of hundreds of times. Specifically, the expressive ability of each trivial template (with only one non-zero element) is weak. Thus a large number of trivial templates are required to compensate for the occlusion appears at any possible locations as an enumeration method [130]. Such a large number of trivial templates working concretely results to high-dimensional coefficients, making the $\ell 1$ optimization expensive.

To reduce the computational cost, some methods try to reduce the dimension of the dictionary $\mathbf{B}$: the Robust Sparse Coding (RSC) model [180] removes the occlusion dictionary $\mathbf{I}$ from the sparse representation. Given this, the only occlusion handling part that RSC can truly rely on is a weighted least absolute shrinkage and selection operator algorithm. From another point of view, some methods try to reduce the number of $\ell 1$ optimization. In [113], only these candidates whose reconstruction errors are above a threshold are selected for $\ell 1$ optimization. Other methods may directly handle the computational issue through the fast numerical method for solving $\ell 1$ optimization [8] or transform this problem into other

ways like solving ℓ_{pq} minimization problem [190]. However, these trackers are still limited when high performance is required for an online tracking task.

(2) Appearance Variation

The appearance variation may be due to the change in the appearance of the target or corruptions act on targets like occlusion and noise. Tracking methods that are not robust enough against appearance variation could lead to drifting as time goes by [185]. Things get worse in sparse coding based trackers which make use of trivial templates $\mathbf{I}$ as variation dictionary. First, the tracker sometimes does not fit the target very accurately. This is not only because that parts of the object may also be represented by the trivial templates, but also because that when updating the appearance model, it is difficult to distinguish between appearance change of the target and interruptions like partial occlusion and noise. Second, the assumption in them that only a small number of pixels are corrupted (in order to use the sparse prior to calculating $\mathbf{c}^*$ in (2.3)) is always not true.

Based on the above issues, the appearance variation should be learned adaptively and be represented precisely. The Bounded Particle Resampling tracker (L1BPR) [113] detects occluded pixels and disable the updating of appearance model when heavy occlusion happens. However, the construction of the variation dictionary is rigid. Since the form of each trivial template is fixed, they cannot adapt to or capture the specialty of various targets. Considering this, [190] represents each candidate by target templates, background templates and error basis, and [191] represents the candidate by target templates, occlusion templates, and context templates. Then the appearance variations can be learned online when new observations are available over time. However, some variations, like long-term occlusions, may appear for a relatively long period of time in consecutive frames [75, 90, 150]. Thus they do contain meaningful prior knowledge for subsequent frames and should not only be regarded as noise. In summary, when modeling the appearance variation, these methods only utilize spatial correlation but leave out the temporal association between the target and corruptions. However, temporal information among consecutive video frames does have key confidential, since the motion information of the target and irrelevant disturbances, including locations, speed, and direction of movements, present different patterns in the video [193]. Thus the expression of corruption could be more comprehensive if

both spatial and temporal information is gathered in building correspondent templates. Apart from this, for sparse coding based methods, no matter appearance variations happen or not, both outdated and representative templates coexist in the dictionary and they are treated equally in the $\ell 1$ optimization process. A tracker with accelerated proximal gradient solver (L1APG) [8] automatically constrains all of the trivial templates when there is no occlusion. However, it cannot limit the interferences coming from inaccurate target templates which may improperly play roles in the appearance model.

2.2 Discriminative Tracking Methods

Discriminative appearance models regard tracking as a binary classification issue and aim to learn some decision bounds between the object and non-object regions discriminately. Discovering highly informative features is important for the discriminative based tracking algorithms. The variants of the tracking target can be incrementally learned with discriminative classifiers for the purpose of object prediction [134]. However, a major limitation of the discriminative based tracking methods is that the appearance model heavily relies on the selection of training samples.

2.2.1 Traditional Methods

For discriminative trackers with hand-crafted features, numerous classifiers have been adapted for object tracking, such as structured support vector machine [64], boosting [159], random forest [73, 137], and discriminative correlation filters [52]. Three typical methods are introduced here.

Boosting-based Methods Boosting-based trackers first train a classifier over the data from previous frames and subsequently use the trained classifier to estimate the location of the target in the current frame. A set of positive samples and negative samples which are labeled by the previously trained classifier, are then selected to update the classifier [58]. D. Tran et al. [91] develops the online GradientBoost which contains a set of noise insensitive loss functions to enhance

the robustness of the tracker. To tackle the drifting problem, transfer learning based methods are developed [53]. D. Levi et al. [174] categorize the samples into the auxiliary sample and target samples and exploring the intrinsic proximity relationships among these samples, leading to robust tracking results.

SVM Based Methods The Support Vector Machine (SVM) based methods discover and remember informative samples as support vectors for the classifier [4, 105], thus has strong discriminative power. [154] use an ensemble of linear SVM classifiers which can be adaptively weighted according to their discriminative abilities in tackling large appearance variations. Then [182] is developed based on a structured output support vector machine. It integrates the structured constraints into the max-margin optimization problem to avoid the heuristic and unreliable step of training sample selection in previous works. In addition, [7] is based on ranking SVM which pose tracking as a weakly supervised ranking problem thus captures the relative proximity relationship between samples towards the true target samples.

Correlation Filter-based Methods Recently, significant attention has been paid to Discriminative Correlation Filters (DCF) based methods for real-time visual tracking. The correlation filter can produce correlation peaks for the target while yield low response to the background, thus can be used as detectors of the target [13, 15, 86, 108]. The general working flow of DCF can be summarized as follows. It is initially trained with image patches cropped around the target in the first frame. Then in the subsequent frames, the candidate patches near/at the previously predicted position is cropped for detection. The correlation operations are performed on these candidate patches and spatial confidence map, or response map, can be obtained. The position with a maximum value in this map is predicted as the new state of the target. Finally, the correlation filter is updated by the appearance at the estimated position. In practice, the correlation procedure is obtained by the inverse Fast Fourier Transform (FFT) operation. According to [21], this workflow can be described mathematically as follows. Suppose $\mathbf{x}$ is either the image patch or extracted features, $\mathbf{h}$ is the correlation filter, and operation$\hat{}$represents the result of FFT. Then the confidence map $\mathbf{M}_c$ is obtained

by

$$\mathbf{M}_c = \mathbf{x} \otimes \mathbf{h} \tag{2.5}$$

$$= \mathcal{F}^{-1}\left(\hat{\mathbf{x}} \odot \hat{\mathbf{h}}^*\right) \tag{2.6}$$

In Eq. 2.6, $*$ represents the complex conjugate of the vector, $\otimes$ denotes the circulant convolution, and $\mathcal{F}^{-1}$ represent the inverse Fourier transform. Suppose $\mathbf{y}$ be a 2D Gaussian shaped labels, then the correlation filter is learned by minimizing the ridge regression objective:

$$\hat{\mathbf{h}}^* = \arg\min_{\hat{\mathbf{h}}^*} \{\Phi(\hat{\mathbf{h}}) = \|\hat{\mathbf{x}} \odot \hat{\mathbf{h}}^* - \hat{\mathbf{y}}\|^2 + \lambda \|\hat{\mathbf{h}}^*\|^2\} \tag{2.7}$$

where λ denotes the regularization parameter. Denote by $\hat{\mathbf{x}}$ the Fourier transform of $\mathbf{x}$, and $\hat{\mathbf{x}}^*$ the complex conjugate of $\mathbf{x}$. With FFT, the closed-form solution to Eq. 2.7 can be given as:

$$\mathbf{h} = \mathcal{F}^{-1}\left(\frac{\hat{\mathbf{x}}^* \otimes \hat{\mathbf{y}}}{\hat{\mathbf{x}}^* \otimes \hat{\mathbf{x}} + \lambda}\right) \tag{2.8}$$

Algorithms based on correlation filtering have demonstrated superior computational efficiency and fairly good tracking accuracy, due to the two important properties. First, by exploiting the properties of circular correlation and performing the correlation operations in the Fourier domain [126, 139], DCFs are suitable for fast tracking. For example, conventional DCF trackers can perform at more than 100 frames per second (FPS) [70], which is significant for real-time trackings. Second, DCFs regress the circularly shifted versions of input features to soft labels, i.e., generated by a Gaussian function ranging from zero to one. In contrast to most existing tracking-by-detection approaches [63, 122] that generate sparse response scores over sampled locations, DCFs always generate dense response scores over all searching locations.

The CF-based trackers can be classified as conventional DCF trackers and trackers combine deep features with correlation filters. We first introduce conventional DCF trackers here. Then the trackers work with deep features are presented in section 2.2.2. The tracker with Minimum Output Sum of Squared

Error filter (MOSSE) [14] encodes target appearance through an adaptive correlation filter by optimizing the output sum of squared error. After the success of MOSSE, several extensions have been proposed to considerably improve tracking accuracy. CSK [70] take the advantage of kernel trick [78, 140] to the correlation filter formula. By exploiting the property of the circulant matrix [61, 69], they provide an efficient solver in the Fourier domain. Then the Kernelized Correlation Filter (KCF) [71] improved the tracking performance by extending the correlation filter to multi-channel inputs and kernel-based training. Other extentions on DCF include subspace learning [101], scale estimation [35], and reliable collection [97]. However, methods based on the DCF usually take a region of interest as the input, which makes it very difficult to exploit the structural information of the target. In addition, the cyclically constructed samples also introduce the unwanted boundary effects. Thus many improvements have also been proposed. Methods like the Spatially Regularized Discriminative Correlation Filters based tracking (SRDCF) [38], Correlation Filters with Limited Boundaries (CFLB) [83], Context-Aware Correlation Filter Tracking (CACF) [118], Background-Aware Correlation Filters for tracking (BACF) [51] are proposed to mitigate boundary effects in the Fourier domain. The better performance is obtained but the high-speed property of DCF is broken. Thus STRCF, an advanced version of SRDCF, incorporates both temporal and spatial regularization to handle boundary effects without much loss in efficiency and achieve superior performance in terms of accuracy and speed. BACF has two limitations: first, it exploits the augmented Lagrangian method for model learning, which limits the model extension; second, even though the background region outside the bounding box is suppressed, the tracker may also be influenced by the background region inside the bounding box. To against the impact from the background, the joint Discrimination and Reliability learning Tracker (DRT) [149] jointly models the discrimination and reliability information. It introduces a local response consistency regular term to emphasize equal contributions from different regions and avoid the tracker being dominated by unreliable regions. Besides, scale adaptive CF trackers are proposed to estimate target scale changes during tracking. The Scale Adaptive with Multiple Features tracker (SAMF) [96] and Discriminative Scale Space Tracker (DSST) [36] are two commonly used methods for scale estimation. In specific,

DSST developed a new correlation filter that can detect scale changes of the target. The tracker searches over the scale space for correlation filters to handle the variation of object size and obtain a good accuracy result in the VOT2015 competition [84]. However, such a strategy is time-consuming in the case of large-scale space, and many improvements over them have been proposed. To speed up scale space searching, the Multi-Kernel Correlation Filter based tracker (MKCF) [151] proposes bi-section search and fast feature scaling method. Then, for more stable detections, the tracker with Multi-Template Scale-Adaptive Kernelized Correlation Filters (MTSAKCF) [11] maximizes the posterior probability rather than the likelihood in different scales. Additionally, Spatio-Temporal Context tracker [186] also suggest a robust scale estimation method that exploits the use of context information to average the scales over several consecutive frames. Despite these successes in isometric scale variation, such kind of methods cannot well address aspect ratio variation. To address this issue, the research in [94] introducing a family of 1D boundary CFs to localize the boundaries in videos and cope with the aspect ratio variation flexibly during tracking. Other efforts in improving the performance of correlation filter based trackers include the Long-term Correlation Tracker (LCT) [107] and the MUlti-Store Tracker (MUSTer) [72] which integrate correlation filters with an additional long-term memory system [3] for long time tracking. The trackers use Attentional Correlation Filter Network (ACFN) [26] and an SCT with four attentional feature-based correlation filters (SCT4) [28] add attentional mechanism exploiting previous target appearance and dynamics into the tracker. Inspired by works like [158] and [144], colour attributes also contributes a lot to DCF based trackers. [40] uses multiple dimensional features, which propose an adaptive low-dimensional variant of colour attributes. Staple tracker [9] combines complementary template and color cues in a ridge regression framework. However, all these DCF-based trackers are developed with handcrafted features, which hinder their accuracy and robustness.

2.2.2 Convolutional Neural Network-based Methods

Since the visual representation is critical for visual tracking, the powerful Convolutional Neural Networks (CNNs) is becoming an ideal feature extractor for this

task. Benefited from the CNN's pre-trained on object recognition and detection tasks [20], visual trackers are more robust to experimental noises. These CNN trackers can be divided into two groups: combine deep features with DCF, or design deep tracking networks.

Combine Deep Features with DCF Since DCF provides an excellent framework for recent tracking research, the first trend is using CNN as a feature extractor and adopt correlation filter as their base tracker. For example, DeepSRDCF [37], an extended work of regularized correlation filter SRDCF, exploits shallow CNN features in a spatially regularized DCF framework. In Hierarchical Convolutional Features based tracking (HCF) [106] and Hedged Deep Tracking (HDT) [131], CNN are employed to extract features which replace handcrafted features, and final tracking results are obtained by combining hierarchical response and hedging weak trackers, respectively. Considering that the CNN's provided features are either coarse and abstract or fine and primitive, HCF proposes to combine feature maps generated by three layers of convolution filters, and introduce a coarse-to-fine searching strategy for target localization. HDT estimates the target position by fusing the response maps obtained from convolutional features of various resolutions. However, the above mentioned DCFs based tracking algorithms are limited by two aspects. First, learning DCFs is independent of feature extraction. Thus the achieved tracking results of these methods may be suboptimal because the chosen CNN features are always pre-trained in different tasks and individual components in tracking systems are learned separately. Second, noisy updates may lead to drifting. This is due to the fact that most DCFs trackers use a linear interpolation operation to update the learned filters over time. Such an empirical interpolation weight is unlikely to strike a good balance between model adaptivity and stability. To overcome these drawbacks, Convolutional RESidual learning for visual Tracking (CREST) [127] reformulates DCFs as a one-layer convolutional neural network that directly generates the response map as the spatial correlation between two consecutive frames. With this formulation, feature extraction through pre-trained CNN models, correlation response map generation, as well as model update are effectively integrated into an end-to-end form. The tracker which uses Continuous Convolution Operators

for visual Tracking (C-COT) [39] employs the implicit interpolation method to solve the learning problem in the continuous spatial domain. As fewer model parameters are used in the model, C-COT is insusceptible to the over-fitting problem. By considering the linear combination of raw deep features, Efficient Convolution Operators for tracking (ECO) [34] is an improved version of C-COT in performance and speed. However, neither C-COT nor ECO are designed towards real-time applications. Then regarding the processing time, TRACA [27] achieves high computational speed over 100 fps. The major contribution to the high computational speed lies in the proposed deep feature compression that is achieved by a context-aware scheme utilizing multiple expert auto-encoders. More recently, Learning Spatial-Aware Regressions for Visual Tracking (LSART) [148] makes a combination of the spatial aware kernelized ridge regression model which focuses on the holistic object, and the spatial-aware CNN model which focuses on small and localized regions. The complementing design results in better performance and is ranked first in performance in the VOT 2017 challenge [85]. These trackers have two major drawbacks. Firstly, they can not end-to-end train and perform tracking systems. Secondly, they only consider appearance features in the current frame and can hardly benefit from motion and inter-frame information. To tackle these issues, some trackers introduce the optical flow as a motion feature for object tracking into the correlation filter. The motion features from different frames provide diverse information for the same object instance, such as different viewpoints, deformation, and varied illuminations. [56] is based on the SRDCF framework, and additionally use deep motion cues to extract discriminative and complementary information that can improve tracking performance. FlowTrack [199] formulate the optical flow estimation, feature extraction, aggregation and correlation filter tracking as special layers in the network, which enables end-to-end learning. Then the previous frames are warped to a specified frame by the guiding of flow information, and they are aggregated for consequent correlation filter tracking. However, they can only tune the hyper-parameters heuristically since feature extraction and tracking process are separated.

Design Deep Tracking Networks The other trend of CNN-based trackers is to design the tracking networks and pre-train them which aim to learn the

target-specific features and handle the challenges for each new video. The deep learning models have become an essential oracle to improve the tracking accuracy, especially for complex tracking scenarios, mainly due to their large model capacities and strong feature learning abilities.

Discriminative model-based trackers first generate multiple target candidates and then refine them with online classification. For example, [163] adopts the use of denoising autoencoder to identify the foreground patch. These methods commonly require a large amount of auxiliary training data as well as off-line pre-training to allow promising performance, which makes the resulting trackers quite computational costly. Besides, an efficient online updating scheme is essential for CNN trackers when handling challenges in the new video. The Sequentially Training Convolutional networks for visual Tracking (STCT) [162] propose a sequential training method for CNNs to effectively transfer pre-trained deep features for online visual tracking. The Fully Convolutional Network based Tracker (FCNT) [161] proposes a two-stream fully convolutional network to capture both general and specific object information for visual tracking. However, its tracking components are still independently, so the performance may be impaired. What is more, the FCNT can only perform at 3 FPS on GPU because of its layers switch mechanism and feature map selection method, which hinder it from real-time applications. Compared with FCNT, the Unified Convolutional networks based Tracker (UCT) [198] introduces peak-versus-noise ratio (PNR) criterion into its updating module, and scale changes are handled efficiently by incorporating a scale branch into the network. The Multi-Domain Network based tracker (MDNet) [120] treated tracking as a classification problem. It trained a multi-domain network, which has shared CNN layers to capture a generic feature representation, and separate branches of subsequent domain-specific layers to do the binary classification (target vs. background) for each sequence. The classifier is updated online by adding some learnable fully-connected layers to perform tracking with the Particle Filter framework [23]. Other works include [31] which tries to construct an appearance model with more pixel details but still need careful training and updating procedure, Action-Decision Network based tracker (ADNet) [184] which suggests a new tracking method using an action decision network which can be trained by a reinforcement learning method with weakly

labelled datasets, and Deep Learning Tracker (DLT) [163] which utilize stacked denoising autoencoder to refine the tracker through online classification.

In brief, the above-mentioned trackers based on online deep learning require frequent fine-tuning of the networks, which is slow and prohibits real-time tracking. In addition, since the labeled training data for tracking is limited, online training a convolution neural network is prone to overfitting, which makes it become a challenging task. In contrast to these CNN based methods that require running back-propagation to online train the network during tracking, the Recurrent Filter Learning based tracker (RFL) [181] discards online training, and instead uses a recurrent network to update the target appearance model with each frame to obtain a faster processing speed. VIsual Tracking via Adversarial Learning (VITAL) [147] overcomes the class imbalance between positive and negative samples by taking advantage of the recent progress in adversarial learning which augments training data to facilitate the deep classifier training.

Some excellent trackers are designed with Siamese network, which receives growing attention due to its two-stream identical structure. It compares two branches' features in the implicitly embedded space, especially for contrastive tasks. For the tracking task, it adopts an alternative approach to target classification which actually trains a similarity function for pairs of images, and regards visual tracking as an instance searching problem. In this case, the target image patch in the first frame is regarded as a query image to search the object in the following frames. SINT [153] formulates visual tracking as a verification problem. It trains a Siamese architecture to learn a metric for online target matching. SiamFC [10] introduces a fully-convolutional Siamese network for visual tracking, which maps an exemplar of the target and a larger search area of the second frame to a response map. This network is trained off-line and evaluated without any online fine-tuning. In a similar structure, [68] achieves tracking by predicting location axis. Another similar framework is called GOTURN, where the motion between successive frames is predicted using a deep regression network. In specific, it is trained to regress the targets position and size directly by inputting the network with a search image (current frame) and a query image (previous frame) that contains the target. However, these mentioned studies only denote tracking result with bounding boxes by providing location and scale

information of the target. It lacks semantic information and inevitably contains corruptions from background areas. Inspired by the contribution in context-aware tracker [42, 60, 172, 197] and the success in detection [133], the work in [24] utilizes the context information with object proposals to build a tracker with finer results. The Siamese region proposal network based tracker (SiamRPN) [93] consists of Siamese subnetwork for feature extraction and region proposal subnetwork including the classification branch and regression branch. It generates a small number of high-quality proposals by a novel instance-specific objectness measure. Different from standard RPN, it uses the correlation feature map of the two branches for proposal extraction. In addition, it does not have pre-defined categories, the template branches used to encode the targets appearance information into the RPN feature map to discriminate foreground from background. Then the Residual Attentional Siamese Network based tracker (RASNet) [166] introduces different kinds of attention mechanisms into the tracking model learning to produce more adaptive discriminative learning. With an end-to-end deep architecture, it extensively explores diverse attentional mechanisms to adapt the offline learned contextualized and multi-scaled feature representation to a specific tracking target. The residual learning within the RASNet further helps to encode a more adaptive representation of the object from multiple levels and a weighted cross correlation layer is proposed to learn the Siamese structure. To guarantee high tracking efficiency, all these learning processes are performed during the offline training stage. Above mentioned methods have superior performance in the speed, which are able to perform at 86 FPS and 100 FPS respectively on GPU. On the one hand, their simplicity and fixed-model nature lead to high speed because no fine-tuning is performed. On the other hand, this also loses the ability to update the appearance model online which is often critical to account for drastic appearance changes in tracking scenarios. Therefore, there still is an improvement space of performance for real-time deep trackers. SA-Siam [65] improves the generalization capability of SiamFC with expressive features and corresponding classifiers that are simultaneously discriminative and generalized. The enhanced generalization capability results in a tracker which is more robust against significant appearance change. Conventionally, both the discrimination and the generalization power need to be strengthened through an online training

process. However, online updating is time-consuming, especially when a large number of parameters are involved. Thus it is crucial for the CNN based trackers to balance the tracking performance and the run-time speed.

2.3 Other Related Research Areas

Our work also has connections with other computer vision fields, such as object detection and video segmentation. Hence, brief reviews on the most related methods of these topics are provided here.

Object Detection The arguably widely used visual detectors are based on sliding-windows, where a scanner exhausts all possible locations on the image plane, and a classifier determines whether a certain region contains the object of interest. Numerious research efforts have been made to improve the performance of a detector, including these dealing with features [33, 136, 169, 183], classifiers [109, 119] , or exploring other effective schemes [43, 171]. In addition, since the scenario of visual detection becomes more challenging, it becomes increasingly difficult for the scanner to explore more general regions to allow flexible presence of complex objects, and limit the total number of regions to be processed by the classifier to keep the system in high efficiency. To alleviate the burden of scanning, efforts have been made, such as shifting some flexibility from the scanner to the classifier [49], or to allow the classifier to reject unlikely regions with fewer computations [173]. Nevertheless, a consensus has been reached that more powerful detectors could be constructed if we can make a wise revision of the practice of blindly exhausting all regions within an image.

Detectors can address the task of classification and where-to-classify simultaneously, utilizing the knowledge of object appearance gained in a learning stage [12, 142]. There are also schemes treat these two problems separately [2, 132, 157, 200]. It is convenient to be able to generate candidate detection windows for generic objects, especially in the case that various categories of objects are of interest and multiple category-specific classifiers are subscribing to a common window proposer. By drawing inspiration from the research in interest points [115], saliency detection [74, 165], and semantic segmentation [55], the

objectness measure exactly applies to this case. It quantifies how likely for a candidate window to cover an object of any class, thus is helpful in producing object proposals. However, generating accurate object proposals is extremely challenging since objects may compose of heterogeneous colors, textures, and shapes [82]. Besides, it may appear at any location in the scene with various sizes. In order to cope with this tough situation, some of the object proposal generators aim at digging more powerful features [200]; some of them increase the number of candidate windows to thousands even millions level, to ensure the coverage of target object; and some focus on delicate framework like cascade [132] or bottom-up grouping [157] to improve the overall performance.

With the development of Deep Learning, the features extracted by CNN layers become more powerful than hand-crafted features, which significantly benefits object detection and other vision applications [62,103,164]. Considering this, Fast R-CNN [54] introduces the object proposals generated by hand-crafted features into the object detection framework. The CNN features are generated and classified based on the proposal areas, and the detection accuracy can be improved. After that, the Faster R-CNN [133] is proposed. It makes use of RPN to generate high-quality proposals with CNN features and largely reduce the number of candidate proposals thus performs more efficiently than Fast R-CNN. RPN predicts which box at each image location may contain an object and then decides how to adjust the predicted box to better cover the object. By checking image contents, it is possible to locate semantically interesting areas and filter out a large number of useless object proposals to reduce computation costs. These proposals are generated by sliding windows; thus target at any location can be detected. However, RPN uses a single convolutional kernel for predicting proposals at each location. As a result, proposals generated at the same location share the same receptive field. Since the same information is used, judging objects with various scales and aspect ratios is not easy. The Single Shot multiBox Detector (SSD) [102], an advanced work of RPN, relieves the scaling issue by generating object proposals at multiple feature maps from the later stages of a network in order to perform detection at multiple scales. However, SSD cannot avoid enumerating bounding boxes as proposals since they are needed to deal with the aspect ratio issue. Besides, if the proposal represented by the bounding box is too small, it is hard to hit

the target. If the proposal area is too large, it will lose precision in localization. In addition, these RPN based methods are not category independent.

Our work in chapter 5 addresses the problem from an alternative angle when comparing it to the existing measurements and proposal generating policies. As aforementioned, the transformation models can be adapted to any object proposal generating framework, and as a result, the overall detection system gains efficiency or accuracy or both with little overheads. The proposed method not only has flexible receptive filed to deal with various scale and aspect ratio but also can be category independent.

Our research is also connected with the works in visual attention, which has been studied via both biological and artificial neuron networks [117, 123, 152]. It is notable that our work is more focus on benefiting a practical object detection system, rather than exploring the biological mechanism of visual attention.

Video Segmentation The tracking results which are depicted in the bounding box format only provide location and scale information of the target. It lacks semantic information and inevitably contains corruptions from background areas. Algorithms based on video segmentation illuminate us about how to acquire a more accurate representation of the target, since these methods output the specific shape of target together with its location. To acquire a more robust performance, most video segmentation methods take both visual and temporal information as input [22]. Compared with single image segmentation, the temporal information is key for capturing the latest stage of a target. For instance, [125] uses unsupervised motion-based segmentation on videos to obtain segments and FusionSeg [77] adapts optical flow as temporal hint. Different from above, the One-Shot Video Object Segmentation (Osvos) [16] do not use any temporal information and process each frame independently as they are uncorrelated. Thus the performance of Osvos is strongly depended on the pre-trained models developed upon millions of images. However, the performance of these segmentation methods is restricted by lacking densely labeled training data. Thus [127] generates artificial masks by deforming the annotated mask via affine transformation as well as non-rigid deformation via thin-plate splines. [77] gets hypothesized foreground regions from bounding boxes to generate training samples. However, a single

object may display multiple motions simultaneously. To learn the rich signals in unconstrained images, sufficient training data is necessary for video segmentation methods.

Our method in chapter 4 is different from one-shot learning based trackers. These trackers employ a quick tunning upon observing the target object, which often dubbed as one-shot learning or appearance model [113] [16]. Our work is also different from zero-shot learning method [192]. Zero-shot needs an intermediate description to extrapolate to novel classes, which is not applicable to tracking.

Chapter 3

Template-Based Tracking with Sparse Representation

This chapter details the research for the template-based tracker called MMST which belongs to the generative method. It enhances the existing appearance model based on sparse representation by introducing mask templates produced by frame difference with efficient solutions as well as including system dynamics in the model. The spatial and temporal context in consecutive frames is encoded and updated by these templates to build a robust appearance model. The proposed tracker is robust against occlusion.

3.1 Introduction

As mentioned in the literature review section (Chapter 2.1), the sparse representation model represents the target with a linear combination of templates in a dictionary, which has been successfully used in appearance modelling for a target. These templates comprise two parts, the target template used to describe the rough appearance of the target and the trivial templates that make a complete and trivial set of bases of the template images. However, the sparse representation model, which uses the trivial template set to represent corruptions as described above, has two intrinsic limitations: high computational complexity and hard to deal with appearance variation. The former is caused by the bottleneck of solving

the $\ell 1$ optimization while the latter is caused by corruptions such as occlusion and noise that are beyond the representation power of trivial templates.

To overcome these two limitations, by taking advantage of the spatial and temporal context, we propose a novel dynamically modulated mask sparse tracking (MMST) method. Taking advantage of the temporal context, a temporal scale pyramid that is composed of a range of frames that conserves the object changes in different time scales, is proposed. The algorithm uses the frame difference produced by the temporal scale pyramid to build mask templates that capture the sudden corruptions on the target. The benefits from the mask sparse representation are as follows. Firstly, it is a kind of *personal tailor* of different tracking objects and corruptions. The non-zero pixels in the mask templates can fit the corruption's shape more precisely than trivial templates. Furthermore, the knowledge of corruptions in the mask templates is preserved in a self-learning mechanism. This learning mechanism can be implemented with frame differences. The learned out-of-target mask templates 1) are complementary to the target ones to capture the spatial information among target and corruptions; and 2) possess a proper structure to record the temporal context among them. In addition, a competitive advantage lies in speeding up the process of solving the ℓ_1 minimization problem: massive trivial templates are replaced with a small number of mask templates while still maintaining the good performance of tracker. We further boost the performance of MMST with the spatial context through dynamically modulated templates to limit the interferences coming from inaccurate templates in the sparse representation stage. In this way, accumulated errors, as well as the drifting phenomenon, are alleviated due to the increase in the probability that candidate samples can exactly cover the target. Lastly, our tracker is highly applicable to practical online tracking usage since we provide an efficient solver for it.

3.2 Sparse Tracking with Mask templates

Compared with traditional appearance models used in sparse representation, the proposed model stands at higher perspective and portrays the target and corruptions separately in different layers like the form of oil painting to depict the

inter and intra relationship between them more precisely. In traditional methods, both the target and the corruption are represented with a flat plain model, in which the corruption is considered as part of object appearance. As shown in Fig. 3.1, when the subject's mouth is occluded by the book, the appearance model will make the book contribute to the representation of the face. This is because the algorithm cannot distinguish between occlusion and the real appearance of the target as mentioned before. However, by simulating human visual perception process and modelling the appearance of objects in a three-dimensional way, the real target (subject's face) will locate in the base layer and the occlusion (book) will stand in an upper layer. In this case, the corruption acts like a mask covered on the real target. It is reasonable to do this. First, corruptions like occlusion and illumination variance normally change in different pace and direction. As Fig. 3.1 shows, the cropped area of frame difference represented by image m_2 naturally contains spatial information of corruptions (the book) with pixel intensity. Second, frame differences with a various number of interval frames (e.g., m_2 and m_5) describe multiple scales of the evolution of the motion of corruptions over a period of time. By synthesizing the temporal information together, we can make relatively good and dynamic predictions of the variance pattern of corruptions and objects, which further helps us to accurately predict the state of the object area in the next frame. Considering above, we tend to construct the mask templates, which are complementary with target templates, based on the on-going changes in the target area of video frame.

In particular, a Temporal Scale Pyramid (TSP) composed of multiple backward windows is used in our scheme. The TSP is built in the "bottom-up" manner with frame difference in multiple temporal scales. Each temporal scale is defined by the time span ς of the backward window, which means that the model subtracts the image patch locates at time point $t-\varsigma$ from the one at $t-1$ to get a mask template at this scale. Since the template-sized tracking result areas of past frames have been gotten under the framework of the particle filter, the frame difference between results at $t-\varsigma$ and $t-1$ can be easily obtained to manage this task. Further, the construction of the mask templates under various scales is shown in Fig. 3.2. The bottom layer of TSP consists of consecutive previous frames from time $k-n$ to $k-1$. In the illustration, we let $n=7$

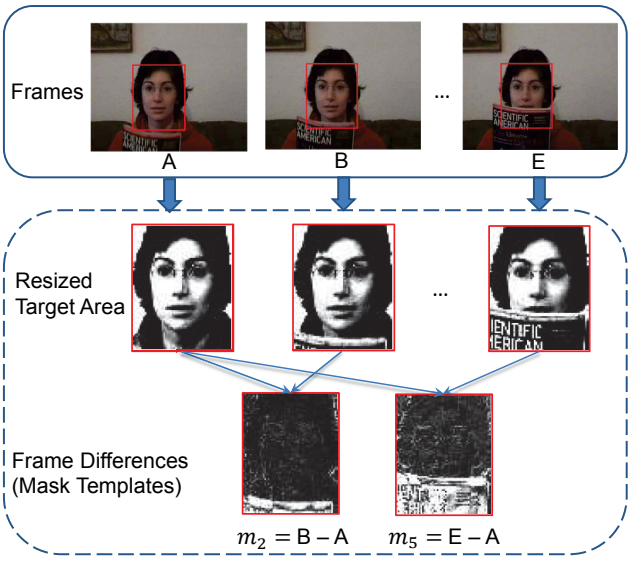


Figure 3.1: Frame differences with a various number of interval frames (e.g. m_2 and m_5) describe multiple scales of the evolution of the motion of corruptions over a period of time.

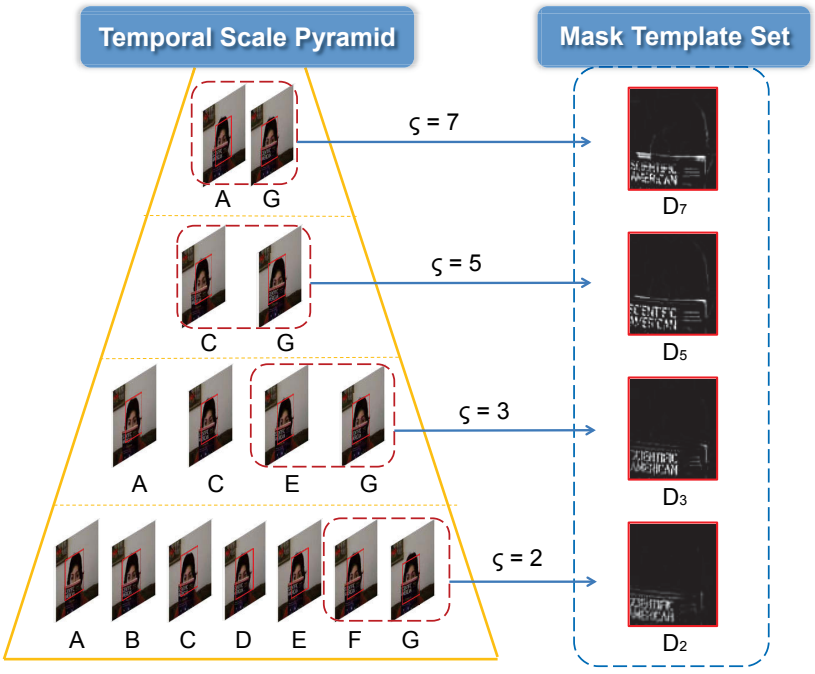


Figure 3.2: Establishing mask templates based on temporal context. The mask template can capture the on-going changes in the target area.

and remark these frames from A to G. The red bounding boxes marked on these frames illustrate the tracking results of frames. We set $\varsigma = 2$ for the bottom layer. Therefore, by calculating frame difference between resized image patches cropped from those bounding boxes in adjacent frames, the minimum changes of the target in a short moment can be precisely depicted in mask template subset marked as D_2. The second layer of TSP is composed of frame images sampled every two frames. Thus frames A, C, E, G are picked out. Also, the patch differences generated by two of them adjacent to each other (correspond to $\varsigma = 3$) are preserved by another subset of mask templates in the same manner as the bottom layer. These mask templates are related to a larger scale of the change of corruptions over time. Similarly, frames C, G lie in the third layer are used to generate subset correspond to $\varsigma = 5$ and A, G in top layer for the subset that $\varsigma = 7$. Thus changes in different scales can be successfully obtained with TSP. In summary, we achieve triple benefits by this model. First, the appearance change, no matter how distinct or inconspicuous, can be accurately detected by mask templates. As shown in the right part of Fig. 3.2, when to compare among layers, it is obvious that the difference among subsets is significant. The subset D_2 precisely records the fast movement of corruption (occlusion from a book) while D_7 marks long-term variation of the target (outline of the target's head). Second, the interferences produced by the movement of the camera or the target are greatly alleviated because of a two-step processing procedure which lies in the mask template construction. Firstly, we aligned the image patches cropped from bounding boxes by the affine transformation. Subsequently, since the corruptions from the background are excluded, we could minus two pure targets easily. Lastly, such scheme saves computational resources not only because just a small patch cropped from one previous frame needs to be stored, but also because the difference is to calculate on an undersized template rather than the whole image frame.

Applying our multi-scale templates to the appearance model in (2.2), the corruptions can be well represented, just like what trivial templates do in traditional sparse coding representation schemes. In our method, the particle observations

can be represented by (3.1).

$$\mathbf{y} = [\mathbf{T}, \mathbf{M}] \begin{bmatrix} \mathbf{a} \\ \mathbf{e_m} \end{bmatrix} \triangleq \mathbf{Bc} \tag{3.1}$$

In (3.1), $\mathbf{T}$ represents target templates and $\mathbf{M}$ contains about a dozen of mask templates. The target templates are constructed with visual observations of the tracked object possibly under a range of appearance changes as in [111]. By image patch subtraction, each mask template acquires rich knowledge of corruptions. Vector $\mathbf{a}$ and $\mathbf{e_m}$ are corresponding coefficients of $\mathbf{T}$ and $\mathbf{M}$. Compared with (2.2), both the dimension of $\ell 1$ minimization problem and the number of mask templates are no longer affected by the size of the template. As a result, it is obvious that $\mathbf{e_m} \ll \mathbf{e}$ can be achieved and the dimension of coefficients is reduced.

For the proposed model, the sparse representation of the target candidate $\mathbf{y}$ is formulated as a minimum error reconstruction issue through a regularized l_1 minimization function with non-negativity constraints as (2.3) describes. The major difference lies in our work is that $\mathbf{B} = [\mathbf{T}, \mathbf{M}]$ is composed of target template set $\mathbf{T}$ and mask template set $\mathbf{M}$. Correspondly, $\mathbf{c} = [\mathbf{a}, \mathbf{e_m}]$, where $\mathbf{e}$ is replaced by mask coefficients $\mathbf{e_m}$. Both $\mathbf{a}$ and $\mathbf{e_m}$ are non-negativity coefficients.

3.3 Performance boosting

In the previous part, we made use of mask templates to tackle sudden corruptions like occlusions and noises. Then we further improve the performance of our tracker, especially in drifting resistance, by considering the system dynamic. In the following part, we introduce how to manage this through dynamic consistency estimation. With this estimation, we could dig into the invariant information among frames, namely dynamic consistency, to promote the performance of our tracker.

It is obvious that, during a short time-slot, some abstract features of an object remain unchanged. For example, when comparing among a few successive frames of a video that records a running car, we can find out that the velocity, angle, and scale of the car may change equably. These attributes whose rate of

change keep constant are marked as *target's state*. Then, an object's state in the next frame can be predicted with the knowledge from former frames. Inspired by the predictable states information, we propose a dynamic consistency estimation mechanism, which mainly overcomes two drawbacks in particle filter based trackers. First, with the assistance from dynamic consistency estimation, the overall performance of particles is more approximate to the real target, whereas in traditional approaches [8,111,112], particles are randomly sampled around the state of the tracked object in the last frame. If an object's state is changing in a roughly predictable direction, it is more reasonable to sample particles based on the predicting state, rather than the state in the last frame. Second, different from traditional methods which treat particles equally, our state estimation system picks out particles that are more similar to the predicted ones. Consequently, the observation likelihood of state $\mathbf{x}_t$ can be calculated by combining reconstruction error and state estimation value that summarized as:

$$p(\mathbf{z}_t|\mathbf{x}_t) = \frac{1}{\Gamma}\exp\{-\alpha\left\|\mathbf{Ta}-\mathbf{y}\right\|_2^2 - \beta\left\|(S_p - S_e)\Psi\right\|_2^2\} \tag{3.2}$$

where the state variable S_p denotes the state of each particle, and S_e represents the predicting state of the target. Both of them can be modeled by six parameters of the affine transformation. Additionally, $\Psi = diag(\mu_1, \ldots, \mu_6)$ are parameters used to normalize these parameters and adjust their significance in our state estimation compensation. If a particle is similar to the predicted state, then the distance between S_p and S_e is small, and this particle will be assigned with a higher observation likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$. With (3.2), we use the prediction to weight particles according to their different states in the process of calculating the possibilities. Therefore, by revising the re-sampling process according to the state prediction, the drifting phenomenon is alleviated due to the fact that candidate samples could cover the target accurately with a higher probability.

3.4 Dynamically Modulated MST with Efficient Solver

With the mask sparse representation model described above, each candidate target patch can be found via solving a minimum error reconstruction problem. However, traditional minimum error reconstruction schemes which could be translated into (2.3) have an obvious defect in common: templates that are less useful in constructing a robust appearance model are treated as equally as the useful ones when chosen as candidates for the component of the appearance model. To tackle this problem, we utilize the spatial context to propose a dynamically modulated mask sparse tracking model which is able to limit the interferences coming from inaccurate templates in sparse representation stage. In our model, an adaptive parameter regularization scheme designed for both target and mask templates is adopted to represent the target more accurately and efficiently, by accounting for occlusion using a specialized regularization scheme. In addition, we demonstrate that the proposed model can be solved more efficiently with APG algorithm.

In model construction, the contribution of each template can be regulated by the regularization parameter which acts on the template's coefficient. Smaller parameters are assigned to more important templates, allowing the template to contribute more significance to the appearance model. Furthermore, the criteria for adjusting regularization parameters are different between target templates and mask templates and are also affected by whether corruptions are detected. The above motivations lead to the following minimization model for dynamically modulated mask sparse tracking:

$$\begin{aligned}\mathbf{c}^* &= \arg\min_{\mathbf{c}} \|\mathbf{B}\mathbf{c}-\mathbf{y}\|_2^2 + \|\mathbf{l}\odot\mathbf{a}\|_1 + \boldsymbol{\mu}\|\mathbf{e_m}\|_1 \\ &= \arg\min_{\mathbf{c}} \|\mathbf{B}\mathbf{c}-\mathbf{y}\|_2^2 + \lambda_1|a_1| + \cdots + \lambda_N|a_N| \\ &\quad + \boldsymbol{\mu}\|\mathbf{e_m}\|_1 \quad s.t. \quad \mathbf{c}\succeq 0\end{aligned} \tag{3.3}$$

where $\mathbf{B} = [\mathbf{T}, \mathbf{M}]$, $\mathbf{c} = [\mathbf{a}, \mathbf{e_m}]$ have the same definition with (3.1). The operator $\odot$ is element wise product. $\mathbf{l} = [\lambda_1, \lambda_2, \cdots, \lambda_N]$ are non-negative regularization parameters adjusting the contribution of target templates while vector

$\boldsymbol{\mu} = [\mu_1, \mu_2, \cdots, \mu_M]$ are non-negative regularization parameters designed for mask templates. The value of $\mathbf{l}$ and $\boldsymbol{\mu}$ are set as follows.

First, the regularization parameters $\mathbf{l}$ are assigned according to the similarity between target templates and the tracked object in previous frames. Due to this, the target templates which preserve correct appearance information of the object are of significance, and the appearance model will be less disturbed by corruptions or out-of-date target templates. Specifically, if a target template is similar to the object, the intersection angles (θ_i) between them is small and the template should be less constrained by λ_i. Thus the corresponding regularization parameter is computed as $\lambda_i = 1 - \beta \cos\theta_i$, where β is a constant in interval $(0, 1)$.

Second, the parameter vector $\boldsymbol{\mu}$ of mask templates are assigned according to the derivation of each template. If produced from the bottom layer of TSP, templates will be less penalized due to the continuity of corruptions, rendering them more important in building the appearance model. In contrast, the penalty for mask templates produced by TSP's upper layers will increase. On the other hand, since they contain crucial information about long-term variation tendency of the appearance model, they have also included in mask template set even though they are not as important as other mask templates in the current reconstruction process.

Third, since mask templates are to compensate for corruptions in the tracking process, they will be attached with higher significance when heavy corruption appears. Hence, when obvious corruptions are detected, the penalty for mask templates $\boldsymbol{\mu} = [\mu_1, \mu_2, \cdots, \mu_M]$ will be alleviated for the sake of endowing them more power to capture the corruptions. Furthermore, in this situation, the similarity between target templates and the target can no longer express the variation tendency of the appearance model. Therefore, to protect the regularization parameters of target templates from the disturbance of heavy corruptions and restrict their energy in the reconstruction process, they are all set to be the same value, which is larger than $\boldsymbol{\mu}$.

The benefits of applying the adaptively regularized parameters are two folds: the target could be represented more accurately and the computational burden is dramatically decreased. For the first benefit, by penalizing the templates with

these regularized parameters, both target and mask templates will be modulated according to their different ability to represent the target. Besides, the energy of the mask template associated coefficients, $\mathbf{e_m}$, can be dynamically adjusted according to the appearance of corruptions. For the second benefit, the length of the coefficients $\mathbf{c}$, which is constant regardless of the size of templates, is much shorter than its counterpart in traditional $\ell 1$ tracking methods, enabling the convergence of coefficient solver with fewer iterations. In addition, our model can be solved with APG approach, a fast numerical method for solving minimization problems proposed in [156]. Next, we will explain how the APG approach is applied in our model seamlessly.

The APG algorithm has been successfully used to solve similar $\ell 1$ tracking models in numerous approaches such as [8] and [189]. Though our model (3.3) looks very different from them, it is actually equivalent in essence to the format which meets the condition of adopting APG algorithm. The demonstration is as follows.

Proof of the Equivalence of Two $\ell 1$-min Problem

Proposition 1. *Minimizing*

$$\begin{aligned}\min_{\mathbf{c}} \|\mathbf{Bc}-\mathbf{y}\|_2^2 + \lambda_1|a_1| + \cdots + \lambda_N|a_N| \\ +\boldsymbol{\mu}\|\mathbf{e_m}\|_1 \quad s.t. \quad \mathbf{c} \succeq 0\end{aligned}$$

can be converted to minimizing

$$\min_{\mathbf{c}} \|\mathbf{Bc}-\mathbf{y}\|_2^2 + \lambda\|\mathbf{c}\|_1 \quad , \quad s.t. \quad \mathbf{c} \succeq 0$$

without increasing computational complexity.

Proof: Suppose the dimension of $\mathbf{m}$ is n. Since $\ell 1$-norm of a vector means

the sum of the absolute values of all of its components, we have:

$$\begin{aligned} & \min_{\mathbf{c}} \|\mathbf{Bc}-\mathbf{y}\|_2^2+\lambda_1|a_1|+\cdots+\lambda_N|a_N| \\ & \quad +\boldsymbol{\mu}\|\mathbf{e_m}\|_1 \quad s.t. \quad \mathbf{c}\succeq 0 \qquad (3.4) \\ = & \min_{\mathbf{c}} \|\mathbf{Bc}-\mathbf{y}\|_2^2+\|\lambda_1 a_1\|_1+\cdots+\|\lambda_N a_N\|_1 \\ & +\|\mu_1 e_{m,1}\|_1+\cdots+\|\mu_M e_{m,M}\|_1 \qquad (3.5) \\ = & \min_{\mathbf{c}} \|\mathbf{Bc}-\mathbf{y}\|_2^2+\|\mathbf{\Lambda c}\|_1 \qquad (3.6) \\ = & \min_{\tilde{\mathbf{c}}} \left\|\mathbf{B\Lambda}^{-1}\tilde{\mathbf{c}}-\mathbf{y}\right\|_2^2+\|\tilde{\mathbf{c}}\|_1 \qquad (3.7) \\ = & \min_{\tilde{\mathbf{c}}} \left\|\tilde{\Phi}\tilde{\mathbf{c}}-\mathbf{y}\right\|_2^2+\|\tilde{\mathbf{c}}\|_1 \quad s.t. \quad \tilde{\mathbf{c}}\succeq 0 \qquad (3.8) \end{aligned}$$

where $\mathbf{\Lambda}=diag(\lambda_1,\cdots,\lambda_N,\mu_1,\cdots,\mu_M)\in\mathbf{R}^{(N+M)\times(N+M)}$ is a diagonal matrix, $\tilde{\mathbf{c}}=\mathbf{\Lambda c}$ and $\tilde{\Phi}=\mathbf{B\Lambda}^{-1}$. Thus proposition 1 is right.

With the help of proposition 1, we can solve (3.3) using APG method as proposed in [8]. We refer readers to [8], [156] for more details of APG method. For further reducing the number of needed l_1 minimizations, we adopt a minimal error bounding method which was proposed in [112] for problems with the same form as (2.3).

The detailed description of the dynamically modulated mask sparse tracker, called MMST, is given in algorithm 1.

3.5 Experimental Results

We implement the proposed method on MATLAB and compare it with 21 state-of-the-art trackers. Among them L1APG [8], MTT [189] are the most similar trackers to our work. Besides, other related methods which use generative model, such as CPF [129], LOT [124], IVT [135] and LSK [99], are also involved. Then, we further compare our method with algorithms which employ discriminate models like OAB [58], SBT [59], MIL [5], CT [187], TLD [81], Struct [64], CSK [70] and CXT [42]. In addition, to make our experiment more objective, we also compare our tracker with other popular trackers like DFT [143], KMS [30], SMS [29],

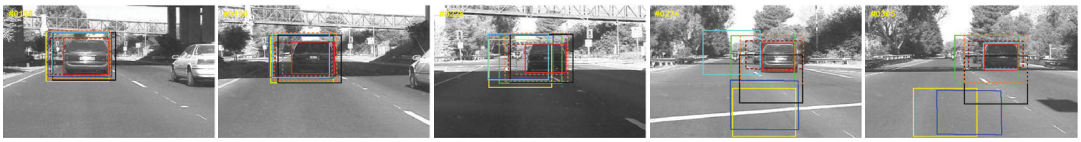
Algorithm 1 MMST Tracker

Input:
Current frame I_t
Sample set $S_t = \{s_t^k\}_{k=1}^M$
Target and mask template sets $\mathbf{T} = \{t_i\}_{i=1}^N$, $\mathbf{M} = \{m_i\}_{i=1}^K$
Compute q_i and τ according to [112]
while $i < M$ and $q_i \geq \tau$ **do**
    Solve the l_1 minimization problem (8) for y_t^i
    Compute the observation likelihood $p(\boldsymbol{z}_t|s_t^i)$ (denoted as p_i for short)
end while
Set $p_j = 0, \forall j \geq i$
Output:
Find the maximum value of p_i and obtain the current tracked result s_t^*
Detect occlusion and update μ_i and λ_i
Update template set $\mathbf{T}$ and $\mathbf{M}$
Resample $\{s_t^k\}_{k=1}^M$ according to $\{p_k\}_{k=1}^M$

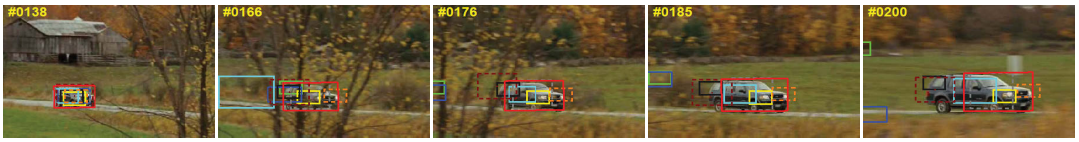
VTD [88], VTS [89], SCM [196] and Frag [1]. To illustrate the effectiveness of our tracker under different tracking conditions, we have gathered 13 video sequences that contain different types of typical corruptions such as occlusion, background clutter, illumination variations, rotation, scale variations, deformation and blurring.

Qualitative comparison and quantitative comparison are carried out in this section. In the Qualitative Comparison part, we present the results of different tracking algorithms in Fig. 3.3 with bounding boxes in various colors. Then in the Quantitative Comparison, we present our results comprehensively in three aspects. First, the details of the performance of the evaluated trackers on selected frames are shown in Fig. 3.4 and Fig. 3.5 by curves. Second, the average performances of the trackers on all the collected video sequences are listed in Table 1 and Table 2. The reader is referred to our website: https://sites.google.com/site/motionbasedmasksparsetracking/ for more detailed results on other sequences. In general, the presented experimental results demonstrate the superiority of our tracker on the average performance and also the robustness against heavy occlusion, rotation, dramatic illumination changes, and scale variations.

Implementation Details The parameters of the 21 methods compared are set according to [176], which is an objective benchmark in the visual tracking area. For each sequence, the initial position of the object obtained from [176] applies for all 22 methods and the standard positions of tracking targets as in [176] are used to calculate three metrics (success rate (SR), center location error (CLE) and overlap rate (OR)) in our experiment. Our algorithm, MMST, is conducted on the following specific settings: ① The number of particles is 800, which is 200 more than the counterpart setting in L1APG (Even though, our algorithm operates faster than L1APG on average as demonstrated in Table 3.3). ② The regularization parameters $[\lambda_1, \lambda_2, \cdots, \lambda_N]$ in (3.3) are initially set equally to 1. During tracking, each of them is computed as $\lambda_i = 1 - \beta \cos\theta_i$, where β is set equal to 0.1. ③ The parameters $\Psi = diag(\mu_1, \ldots, \mu_6)$ used to normalize six parameters of the affine transformation and change their significance in our state estimation compensation are set in two steps. First, set $\mu_1, \ldots, \mu_6$ to be the reciprocal of the six affine parameters of the first frame. Second, times $\mu_1, \ldots, \mu_6$ with their own weights set in the dynamic consistency estimation system, which are [0.001, 0.001, 0.001, 0.001, 1, 1].

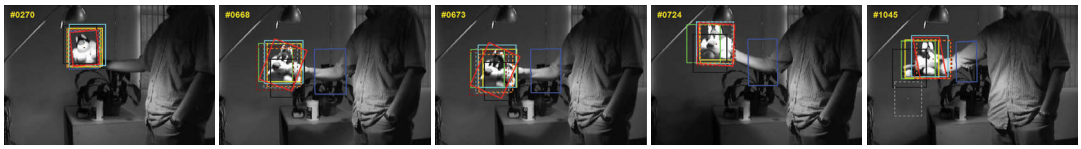


(a) Tracking results of the *car4* sequence

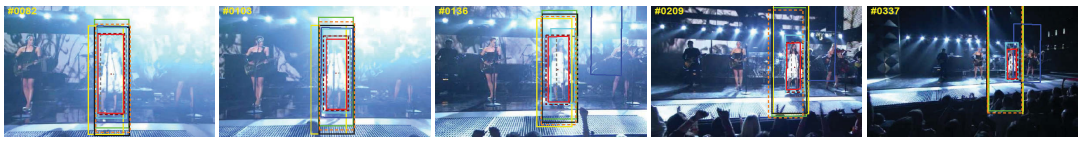


(b) Tracking results of the *carScale* sequence

(c) Tracking results of the *coke* sequence

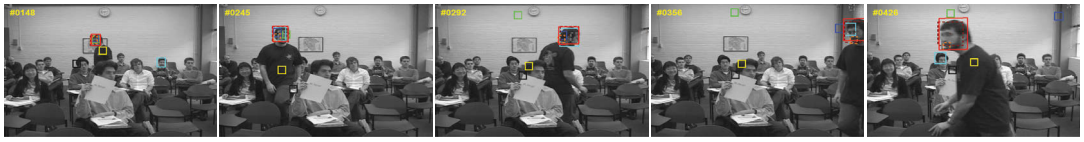


(d) Tracking results of the *sylv* sequence

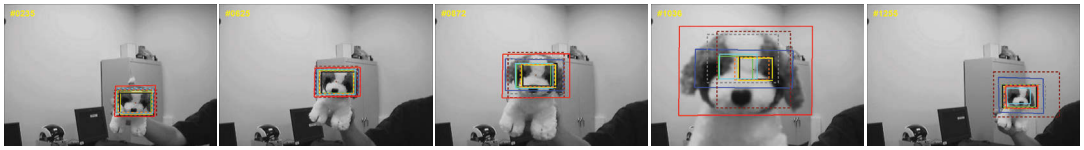


(e) Tracking results of the *singer1* sequence

(f) Tracking results of the *walking2* sequence



(g) Tracking results of the *freeman3* sequence



(h) Tracking results of the *dog1* sequence

CSK CT L1APG MIL TLD IVT SCM Struck MMST

Figure 3.3: Tracking results of different methods on parts of the selected sequences.

Qualitative Comparison In the frames presented in Figure 3.3 , the results of different tracking algorithms are marked by bounding boxes of different colors. To exhibit the performance of different trackers more concise and make them easy to compare, we show the tracking results of 8 representative trackers on 8 sequences.

Occlusion: Representative sequences are *carScale*, *coke* and *walking2*. In the middle of the sequence *carScale*, the car is partially occluded by a tree, making it difficult for a tracker to catch the object precisely. According to the results presented in Fig. 3.3 (b), our tracker is still able to track the target robustly when the car is occluded while other trackers fail to provide an appropriate estimation of the target. In *walking2*, while walking through a corridor, the woman is heavily occluded by a man walking across her from frame #185 to frame #240. When many trackers lose the target and mistakenly regard the man as the target, our tracker is not distracted by the occlusions. A similar situation also happens in the sequence *coke*, when the coke bin is totally blocked by the leaf.

Drifting: The results of drifting resistance are shown in sequence *freeman3*. In Fig. 3.3 (g), the man wearing a T-shirt walks from the center of the room to the right wall and then turn back to the front between frame #0292 and #0426. During this period, L1APG, the tracker which is most similar to our work and marked in purple, drifts to the corner near the right wall and lost the target while MMST keeps tracking of the target successfully.

Appearance Variation and Rotation: Trackers' robustness to rotation and appearance variations can be evaluated in sequences *sylv* and *mhyang*. In *sylv*, the toy mainly undergoes dramatic rotation and appearance variations. It can be found that even in the challenging frames #668, #673, #1045, our tracker can better adapt to the changing appearance of the target. Additionally, at frame #724, our tracker performs favorably even though the illumination changes significantly.

Illumination Variation: The most representative sequences of heavy illumination variation are *car4* and *singer1*. At frame #228 of sequence *car4*, our method is the only one that can still accurately track the target during dramatic illumination variations. In sequence *singer1*, the illumination changes are so dramatic that locating the target is quite challenging even for human beings, rendering

many other trackers drifting from frame #82. Only SCM tracker and our tracker can track the target through the sequence.

Scale Variation: This kind of corruption can be densely shown in sequence *freeman3* and *dog1*. In *freeman3*, since the size of the target is small at the beginning, some information of the background is introduced into the appearance model and disturbs the tracker heavily in successive frames. Frame #148 shows how trackers like TLD, CT, and MIL lose the target when clutter background appears. Moreover, when the man is walking towards the camera, a long-term scale variation and continual rotation happen to his face, enhancing the difficulty of tracking. Then at frame #356, only SCM and our tracker are able to get the target successfully. For sequence *dog1*, due to scale variation and blurring, all the trackers but ours lose the target at frame #1036. Theoretically, SCM, IVT, and L1APG have the potential ability to handle scale variations, but they are not robustness enough when scale variations are accomplished by blurring and rotation, rendering their failure in tracking the dog in this sequence.

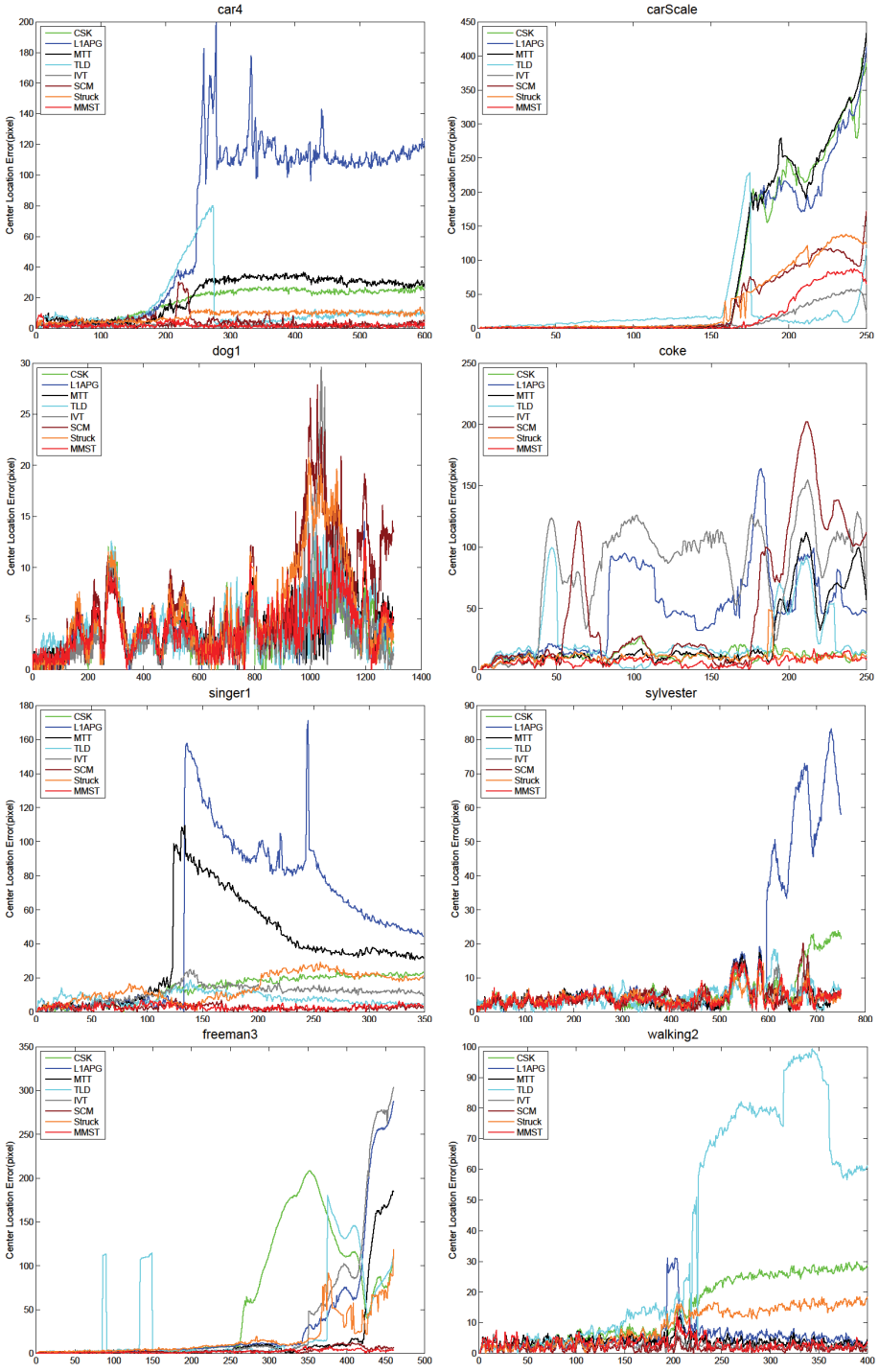


Figure 3.4: Center location error for each test sequence. The result of MMST, which is marked by red lines, has the lower error rate on average for the test sequences.

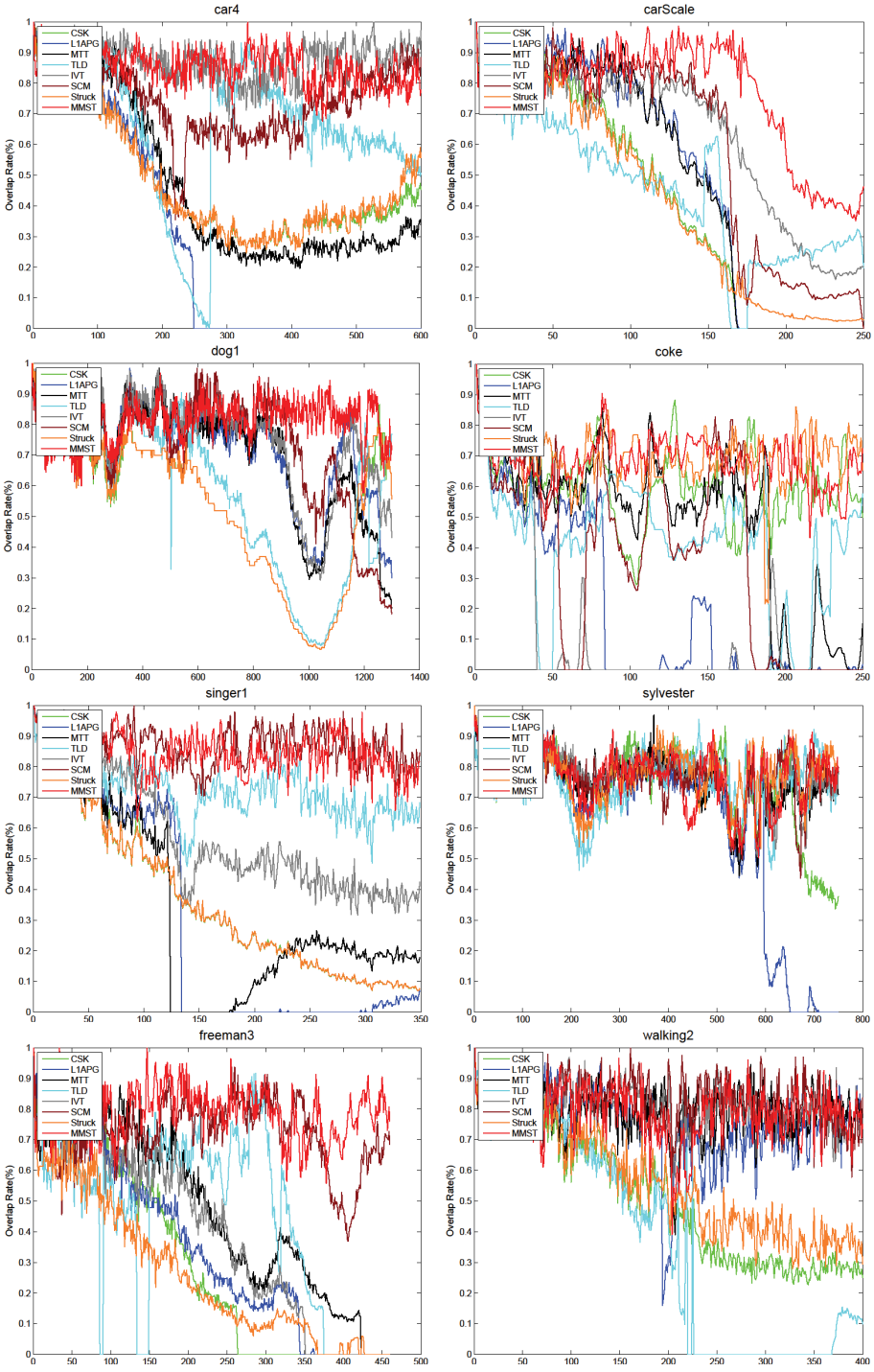


Figure 3.5: Overlap rate for each test sequence. The result of MMST is marked by red lines and has higher overlap rate on average.

Quantitative Comparison Three metrics are adopted to quantitatively evaluate the robustness of our tracker under challenging conditions and compare it to other popular trackers. Following the benchmark [176], we use success rate (SR) and center location error (CLE) to measure the performance. Besides, we also introduce the overlap rate (OR) to the evaluation criterion in order to more precisely judge the similarity in size and shape between tracking results which are marked by bounding boxes and the manually labeled ground truths. Specifically, the OR is quantified as the score in (3.9):

$$S = \frac{area(R_T \bigcap R_G)}{area(R_T \bigcup R_G)} \tag{3.9}$$

where R_T is the region of tracking bounding box and R_G is the region of ground truth bounding box. The SR is computed as the proportion of *success* frames to whole frames of a video. In this metric, the tracking result is considered as *success* if the score S, which is referred to in calculating OR, is large than 0.5 in one frame. The CLE (in pixels) is measured by the Euclidean distance between the center locations of the tracking results and the ground truths.

The details of the performance of each tracker are shown with curves in Fig. 3.4 and Fig. 3.5 which exhibit the CLE and OR result respectively. It can be seen in Fig. 3.4 that in most frames the curve of our method is below others, which indicates that our algorithm can locate the target better than other algorithms. On the other hand, Fig. 3.5 shows our method's outstanding ability to handle the scale variations due to the higher overlap rate of our tracker against other evaluated trackers. For example in *dog1* and *freeman3*, as the object becomes larger, the overlap rate of other methods decrease dramatically compared to our method. Although in some overlap rate curves of sequences such as *crossing* and *faceocc1*, our performance is not the best all the time, the average performance is always close to the top. Moreover, if we synthetically consider both the performances of CLE and OE, our algorithm is better on average. For example, from frame 430 to 450 in the freeman3 sequence, the values of CLE are similar between our tracker and SCM (see Fig. 3.4). However, at the same time, the performance of our tracker in OR is much better than SCM (see Fig. 3.5). Apart from Fig. 3.4 and Fig. 3.5, more detailed results of CLE and OR of trackers on other sequences

Table 3.1: Center location error (in pixels). The blod and italic numbers indicate the best and the second-best respectively.

Method	car4	david2	sylv	mhyang	singer1	coke	crossing	dog1	walking2	walking	freeman3	carScale	faceocc1	Ave
MMST	**2**	**1**	**5**	*3*	**3**	**7**	**2**	**4**	*3*	**2**	**2**	17	15	**5**
CT	84	77	6	13	16	38	3	7	32	7	65	25	26	31
CSK	18	*2*	9	*3*	14	12	9	**4**	15	7	54	81	12	18
DFT	55	17	30	8	19	71	20	37	22	6	33	73	22	32
L1APG	72	**1**	24	**2**	53	49	60	**4**	5	3	33	77	17	31
LOT	158	4	10	111	142	62	34	*5*	45	**2**	41	99	35	58
MTT	21	*2*	**4**	3	36	25	54	**4**	4	3	16	85	21	21
TLD	13	5	5	9	8	24	22	**4**	39	10	29	22	27	17
IVT	**2**	**1**	20	**2**	11	83	**2**	**4**	3	**2**	36	**12**	18	15
SCM	*4*	3	8	**2**	**3**	49	**2**	7	**2**	**2**	*3*	32	13	10
CPF	40	5	10	13	7	43	9	8	48	4	103	31	28	27
Struck	8	*2*	*5*	**2**	14	10	3	6	9	4	17	35	19	10
MIL	49	11	14	19	16	44	3	8	35	3	88	33	30	27
Frag	124	57	14	13	89	119	37	12	47	9	40	19	**11**	45
OAB	88	34	14	7	13	29	4	6	31	5	40	30	25	25
SBT	47	10	62	9	98	48	3	12	12	101	59	27	23	39
KMS	56	35	16	20	53	48	8	23	48	8	64	40	19	34
SMS	131	60	14	15	9	76	8	55	35	8	34	23	23	38
LSK	69	18	63	3	21	49	54	7	22	31	39	13	30	32
VTS	36	3	6	4	5	62	41	12	34	5	18	35	21	22
VTD	36	3	6	4	4	69	25	11	35	6	24	37	20	22
CXT	46	**1**	8	4	11	22	21	*5*	31	198	4	23	25	31

are provided on our website mentioned above.

Besides, in Fig. 3.6 and Fig. 3.7, the overall performance of all the evaluated trackers are illustrated by bars and points. The statistic results with respect to the 22 trackers are marked with different colors, and the relations between the bars and the trackers are listed below the plots. These statistic results are generated by gathering and sorting the CLE and OR results of a tracker on each frame in ascending order. The intermediate 85 percent of tracking results of a tracker on a sequence is represented by the corresponding bar while the remaining 15 percent of results are scattered as points in the top and bottom of the bar. As shown in the plots, the bars of our methods are lower in the CLE and higher in the OR in comparison with other methods, indicating that the average performance of our method is more favorable than other trackers. For example, on sequence *car4*, *carScale* and *freeman3* in Fig. 3.7, the bars of our tracker are generally higher than other trackers. Although our tracker may not achieve the highest bars in some sequences, such as *david2*, *singer1* and *faceocc1*, the difference between our results and the best results is quite small. Moreover, the scattered points of our tracker are closer to the bars when compared to most of the other trackers, which

Table 3.2: Successful rate (in pixels). The blod and italic numbers indicate the best and the second-best respectively.

Method	car4	david2	sylv	mhyang	singer1	coke	crossing	dog1	walking2	walking	freeman3	carScale	faceocc1	Ave
MMST	**100**	**100**	**98**	**100**	**100**	*98*	**100**	**100**	**100**	**100**	**100**	**82**	**100**	**98**
CT	30	0	93	73	25	10	**100**	64	48	52	0	45	85	48
CSK	30	**100**	73	**100**	30	82	35	64	49	54	33	45	**100**	61
DFT	29	55	49	83	28	10	70	54	48	57	33	45	84	50
L1APG	33	**100**	52	*98*	38	23	27	88	97	**100**	33	59	**100**	65
LOT	5	78	73	27	24	32	34	*99*	49	*97*	7	47	31	46
MTT	34	**100**	97	100	35	39	25	82	*99*	96	48	57	**100**	70
TLD	85	95	96	95	*99*	70	56	66	43	40	58	44	83	71
IVT	**100**	93	81	**100**	48	15	26	89	**100**	**100**	44	71	*97*	74
SCM	*97*	91	89	**100**	**100**	39	**100**	88	100	96	93	66	**100**	*89*
CPF	3	46	76	19	32	8	60	97	45	92	15	53	55	46
Struck	34	**100**	97	**100**	30	**99**	*98*	64	54	58	20	44	**100**	69
MIL	30	33	61	41	28	12	**100**	64	48	56	1	45	77	46
Frag	23	30	72	71	22	4	42	61	44	53	31	44	**100**	46
OAB	30	26	74	96	24	20	85	64	48	50	19	44	91	51
SBT	26	55	50	79	18	17	88	61	47	29	20	44	91	48
KMS	25	37	51	54	24	10	55	50	45	52	8	30	94	41
SMS	0	1	1	58	62	4	30	5	47	44	27	54	84	32
LSK	6	64	31	**100**	20	19	13	91	56	61	33	64	43	46
VTS	39	*99*	97	97	43	16	44	68	51	84	34	49	91	62
VTD	39	*99*	97	95	43	15	45	69	51	82	35	48	95	62
CXT	33	**100**	89	88	32	66	37	**100**	50	23	*94*	*79*	79	67

Table 3.3: Speed (fps). Blod fonts indicate the best performance algorithm.

Method	car4	david2	sylv	mhyang	singer1	coke	crossing	dog1	walking2	walking	freeman3	carScale	faceocc1	Ave
MMST	19.82	12.94	14.23	14.96	12.21	13.74	11.5	13.15	14.8	16.38	9.85	14.12	18.44	14.32
SCM	0.49	0.47	0.41	0.52	0.31	0.32	0.38	0.32	0.38	0.35	0.37	0.31	0.43	0.39
L1APG	1.67	2.49	2.2	2.67	2.25	1.32	2.16	3.73	3.3	4.34	1.87	2.6	4.56	2.70
LOT	0.63	0.65	0.56	0.65	0.18	0.23	0.91	0.28	0.3	0.39	0.69	0.37	0.30	0.47

means that our method is more stable on these sequences. In particular, as shown in the *freeman3* of Fig. 3.6, our method has the least scattered points among the evaluated trackers.

Further, the average performances are shown in Table 3.1 and Table 3.2 which exhibit the CLE and SR of 22 trackers on 13 challenging sequences. For CLE, among all the 13 sequences shown in Table 3.1, our approach achieves best performance (marked as bold) regarding half of the sequences. For SR in Table 3.2, our tracker performs the best in all but one sequence. Compared to different sequences, our tracker acquires excellent results in the following situations. First, when the image sequence has corruptions such as heavy occlusion and illumination changes like in *car4, singer1, carScale, coke*. The center location error of our method is significantly reduced in comparison with other template based sparse tracking methods like L1APG. Compared with other template based sparse tracking methods like L1APG, the center location error of our method is significantly reduced. This result validates the effectiveness of employing the mask template set to represent heavy occlusion and illumination changes. Second, the proposed method also performs excellent tracking capability when the target suffering other corruptions like large scale variations (*car4, dog1, freeman3*) and in-plane rotations (*sylv, david2*). Last but not the least, our tracker offers good comprehensive performances at low cost. On average, our method performs much better than other tested methods, especially for similar methods like L1APG.

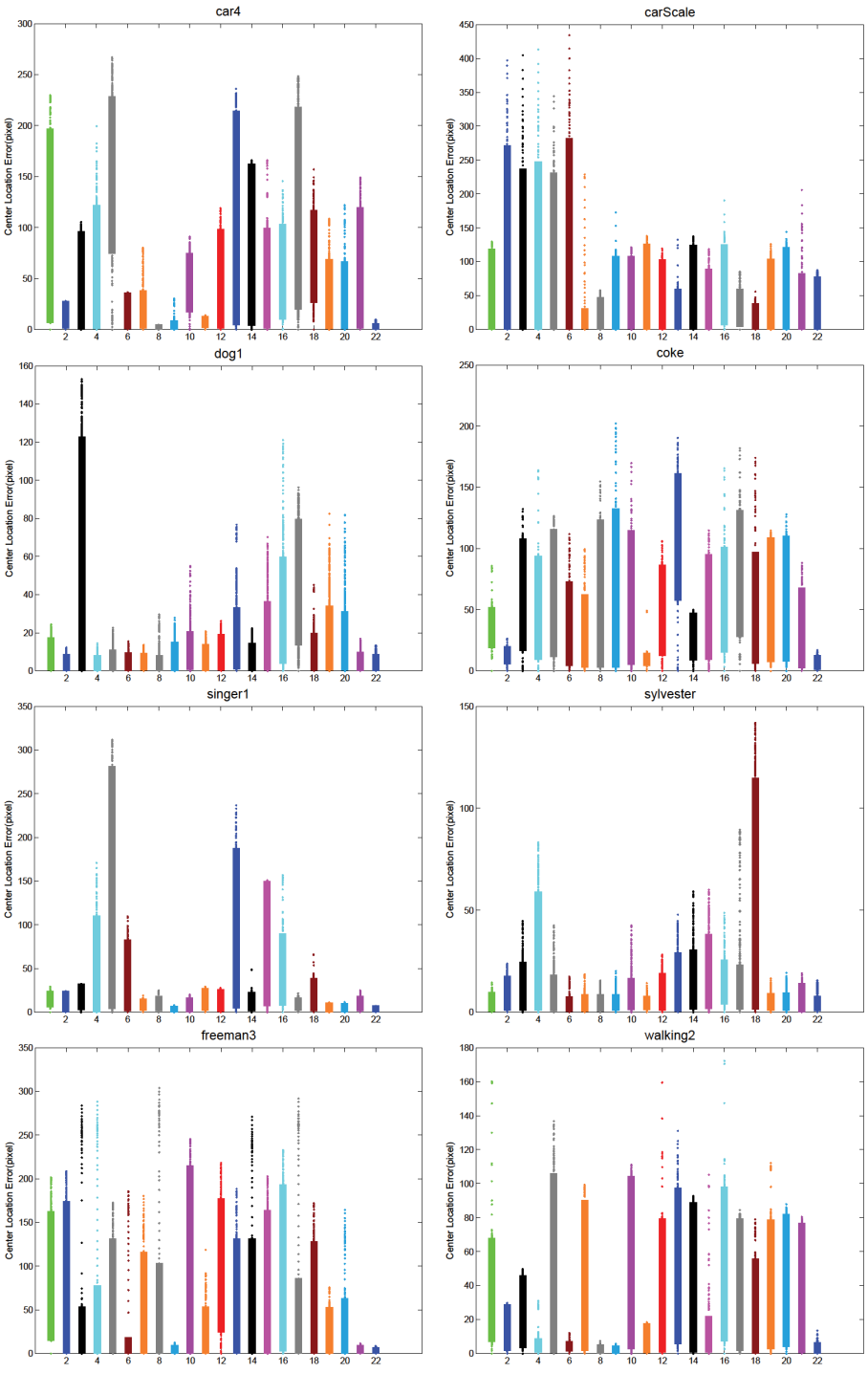
In addition, the comparisons of speed are shown in Table 3.3. The experiment is conducted on a PC with Intel I5-4570 (3.20GHz) and on average, our tracker is about 5 times faster than L1APG, 37 times faster than SCM and 30 times faster than LOT. We choose them because L1APG is the one that mostly related to our method, SCM ranked the second-best in both Table 3.1 and Table 3.2, and LOT is a representative example of the generative method based tracking algorithms. Through the comparisons, we noticed that the speed is heavily related to two facts: First, the number of templates. The length of the related coefficient vector grows as the number of templates increases. This further significantly augments the computational burden for convergence because the goal of $\ell 1$ minimization problem is to find a set of optimized coefficients which lead to the minimum of the equation. In the experiment, for *car4* sequence (similar for other sequences), we

formed our dictionary with only 10 dynamic updated target templates, 1 fixed target templates from the first frame and 4 mask templates. Meanwhile, the original L1APG algorithm uses 10 dynamic updated target templates, 1 fixed target templates and 180 trivial templates, which is about 13 times larger than ours; Second, if the tracking result is corrupted, the speed will seriously decrease. This is because if the templates cannot accurately represent the target, it will take more loops to reach convergence. As a consequence, because our dynamic modulated representation model can represent the result more efficiently and accurately, it will take our model fewer loops to reach convergence than the original L1APG algorithm.

To sum up, considering both the speed and tracking accuracy, our method outperforms 21 other cutting-edge algorithms, especially when the corruption exists.

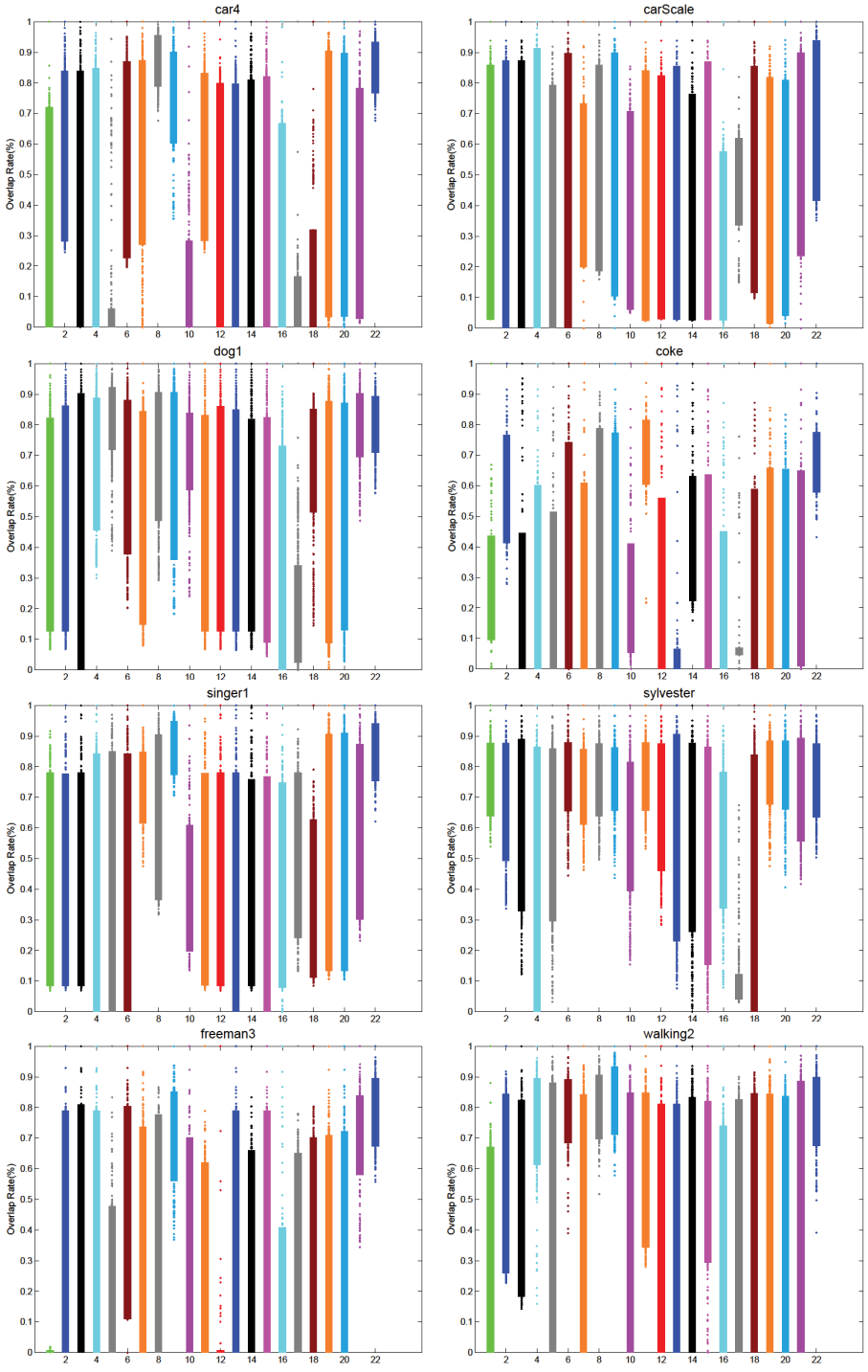
3.6 Conclusion

In this chapter, a robust and efficient visual object tracking method is developed based on an improved subspace learning-based appearance model based on the spatial and temporal context. In our work, mask templates are introduced, significantly helping to reduce the complexity of the system and with a theoretical guarantee of the efficiency of the solution. Our method is also characterized by its exploitation of the dynamic information of the tracking target which significantly improves the tracking accuracy and coverage of the target. Extensive experiments validate the efficiency and robustness of our method, especially in situations with frequent and obvious large-scale corruptions such as occlusions and illumination variations. In future work, our MMST model could be extended to multi-object tracking problems with other promising corruption modelling methods.



1	2	3	4	5	6	7	8	9	10	11
CT	CSK	DFT	L1APG	LOT	MTT	TLD	IVT	SCM	CPF	Struck
12	13	14	15	16	17	18	19	20	21	22
MIL	Frag	OAB	SBT	KMS	SMS	LSK	VTS	VTD	CXT	***MMST***

Figure 3.6: Statistic results of center location error of all trackers.



1	2	3	4	5	6	7	8	9	10	11
CT	CSK	DFT	L1APG	LOT	MTT	TLD	IVT	SCM	CPF	Struck
12	13	14	15	16	17	18	19	20	21	22
MIL	Frag	OAB	SBT	KMS	SMS	LSK	VTS	VTD	CXT	***MMST***

Figure 3.7: Statistic results of overlap rate of all trackers.

Chapter 4

A Robust Tracker Based on a Bi-channel Fully Convolutional Neural Network

This section introduces a bi-channel fully convolutional neural network to tackle a pixel-level tracking problem. The proposed model accepts two video frames as well as the tracking result of the previous frame as input. It has two branches of the sub-network, which can capture and analyse low-level motion variance and high-level semantic variance, respectively. The low-level branch focuses on the temporal context, the movements of local parts of the target across frames, by extracting and operating optical flow data, while the high-level semantic branch concentrates on the spatial-temporal context and outputs the prediction of to-and-fro alternation between background and target for each pixel in the current frame. Both branches employ fully convolutional neural networks for processing. Combining these two, the foreground target area is obtained and can be calculated to carry on the tracking operation for new frames.

4.1 Introduction

Practical object tracking in videos is often formulated as updating the location and size of a bounding box upon observing each new frame in the video, where

the target is specified by the bounding box in the previous frame. Using the bounding box in tracking follows the conventional usage of a rectangular region of interest (ROI). A rectangle is a minimalistic and practical representation of a target and has been ubiquitously used in many machine vision tasks, including object detection [133] and action recognition [155]. On the other hand, pixel-level analytics has long been considered desirable as it provides richer details and naturally accommodates complicated cases such as multi-target detection/tracking, especially when dealing with occlusion and shape variation. Unfortunately, pixel-level processing of images and videos entails the formidable task of capturing fine structures in the visual signals.

A breakthrough has been made recently with the impressive development of deep convolutional neural networks [32, 104]. Given sufficient data and with the cost of an expensive training session, when deployed, these models are able to make quick and accurate predictions at a similar resolution to the input signal [92]. The various receptive fields of CNN kernels, together with a multi-layer architecture which incorporates features at different levels, naturally encode the spatial context to provide a finer result. Thus, a wide range of machine vision tasks, such as object identification and semantic scene understanding, has advanced their granularity of analysis to the pixel level. The work presented in this chapter aims to harvest the benefit of the analytic tools based on neural networks and achieve finer and more accurate object tracking. Although there exists some work dealing with the pixel-level tracking task, such as Osvos [16] and Pixeltrack [45], they always need to fine-tune the model of the trained tracker with the first frame. With this fine-tuning step, the model will be adjusted to better capture the newly come features of the target. In contrast, our method avoids such fine-tuning step which is time-consuming. Our model automatically obtains these features by analysing the temporal context between the newly come frame and the first frame.

4.2 Generic Pixel Level Tracker

The aim is to build a category-independent model to track targets given at run time. In particular, we capture the low-level motion variance to provide an intu-

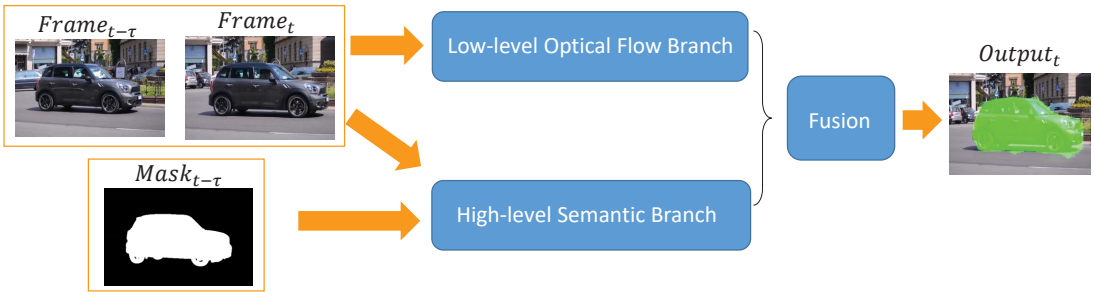


Figure 4.1: The processing flow of the bi-channel fully convolution neural network. Based on the input information, low-level and high-level temporal information are extracted and analysed in corresponding branches. By fusing the results of two branches, the foreground area of the target can be identified.

itive estimation of the movement of each local part of the target, and represent the overall change of the distribution of foreground pixels by introducing high-level target-specific semantic variance. Thus we introduce a bi-channel neural network to process both of the variances for producing a pixel-level tracking result. In particular, the network consists of two processing branches: one for robust prediction of low-level optical flow and the other for tracking high-level semantic objects. Both branches employ the deep fully convolutional network (FCN) architecture [104]. Figure 4.1 shows the structure of the network. The low- and high-level branches share the input of a pair of consecutive video frames, with the high-level branch additionally taking the target object mask in the previous frame. Then after a series of convolutional and de-convolutional feedforward operations, the high-level semantic branch outputs the predicted target object mask in the new frame. The prediction is enhanced by fusing information from the low-level branch, which outputs predicted optical flow summarised in super-pixels by clustering.

Low-level Optical Flow Branch We define that the low-level motion variance represents the displacements of the same pixel in two adjacent frames. Particularly, optical flow is an ideal description of such temporal context variance since a flow of light and colors directly indicates the low-level visual changes of a moving target in the video. However, the raw flow data cannot be directly used to predict the mask of a tracking target, due to the following limitations. First,

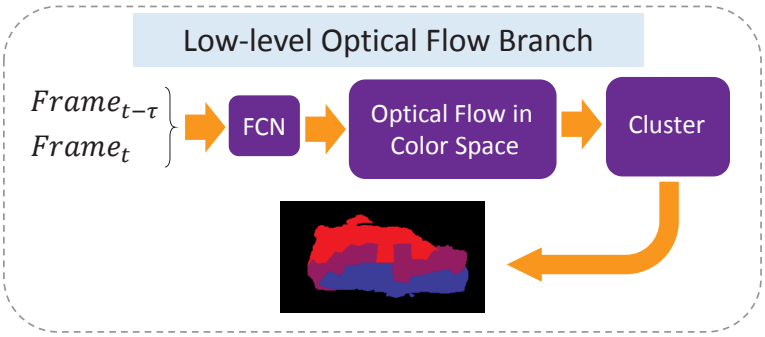


Figure 4.2: The working flow of the low-level branch: the optical flow data is extracted by a fully convolutional neural network with a clustering operation afterwards, so that foreground and background areas can be separated.

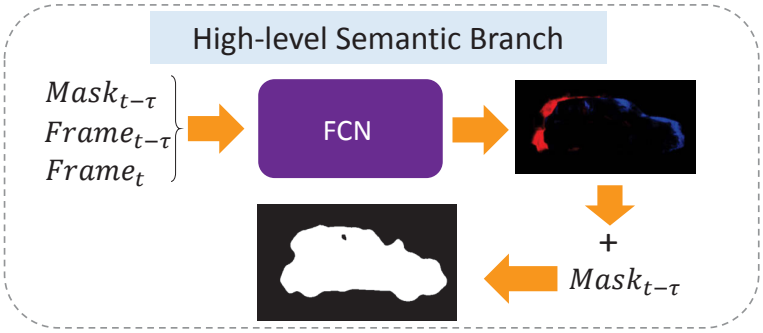


Figure 4.3: The working flow of the high-level branch. It adopts the fully convolutional neural network to predict the decrease and increase (red and blue) of the foreground mask of the target. By adding the predictions to the previous foreground mask, an initial estimation of the target can be obtained.

the raw flow contains noise from the background and would be scattered when corruptions like occlusion appear on the image. Second, when one object moves in diversified speeds and directions, the raw flow will present different features and may confuse the judgment of the algorithm. Third, different parts of a single object may present utterly different optical flow features.

Considering above, we design the low-level branch to extract and manage the optical flow for getting an output where the foreground and background areas are distinguished from each other. To accomplish this, we first refer to a deep convolutional neural network based on FCN to extract the optical flow considering the high speed and accuracy. The network has a similar structure with FlowNetC and FlowNetS provided by Flownet [44]. The number of channels is reduced to make a trade-off for better time efficiency. After that, this branch would process the obtained optical flow using the following steps. Step1: the flow data represented by angle and amplitude are mapped into color images. Step2: optical flows with different attributes (angle, amplitude) are clustered into superpixels, so that the underlying correspondence between flow data and the target can be revealed. Step3: optical flows clustered by the frames at different time intervals are combined, to reduce the impact of variance in moving speed. Figure 4.2 illustrates this process of generating the optical flow summarized in groups by clustering.

High-level Semantic Branch In the high-level branch, we introduce the fully convolutional neural network to update the parsing of object / scene semantics in each new frame regardless of its category. We call this responsible sub-network as "semantic branch".

Mathematically, suppose $M_{t-\tau}$ and M_t are foreground areas at time $t-\tau$ and t respectively. For a pixel located at (x, y), the related semantic variance during time interval τ is marked as $d_{x,y,\tau}$. Then the relationship of $d_{x,y,\tau}$ and M_t can then be written as:

$$M_{x,y,t} = f(M_{x,y,t-\tau} + d_{x,y,\tau}) \tag{4.1}$$

where f is the operation that constrains the values of the changed foreground pixels to lie in $[0, 1]$.

In this branch, we introduce a deep convolutional neural network to directly capture the difference between M_t and $M_{t-\tau}$. Unlike segmentation based algorithms which need prior knowledge as a reference to the foreground area, the proposed network does not need fine-tuning on the first frame to learn the target's appearance from zero. The detailed design of this branch is shown in Fig. 4.3. The inputs include consecutive video frames and tracking results on previous frames. The former contains rich difference information while the latter gives a reference to the location of the target. Three kinds of pixel-level labels ($0, 1$, and 2) are designed for the network to reflect what happens between input images (colored in red, black and blue in Fig. 4.3. If the target mask covers one pixel in the former image but excludes the pixel at the same location in the latter image, label 0 is assigned to the pixel to represent target vanish on it. On the contrary, label 2 will be assigned to such a pixel which is newly added to the target mask in latter images. Label 1 covers the remaining situations: the attribution of the pixel does not change during the interval between images. It remains to the target or background during the time-slot. The basic architecture of the neural network is based on FCN [104] except that batch normalization is introduced to stabilize the training procedure. In addition, to capture more details about the variance, the feature maps are upsampled to the input image size. Furthermore, multiple image pairs of different time intervals are loaded to better capture the change. The branch generates a foreground probability map at last.

Fusion Based on the observation that the outputs two branches share locations on the image, the output of high-level semantic branch can be directly enriched with flow data at the same location given by the low-level optical flow branch. By fusing the outputs of two branches, we obtain the appropriate tracking results.

The detailed algorithm can be summarized using a four-stage procedure. In the first stage, we perform a voting scheme on the optical flow in groups according to the foreground probabilities at the shared location. In the second stage, we distinguish out foreground clusters and background clusters based on a threshold, with an appearance descriptor constructed for each group. In this work, the appearance descriptor is the average value of the attribute of corresponding optical flow. Then the third stage discards the foreground areas predicted in stage 1

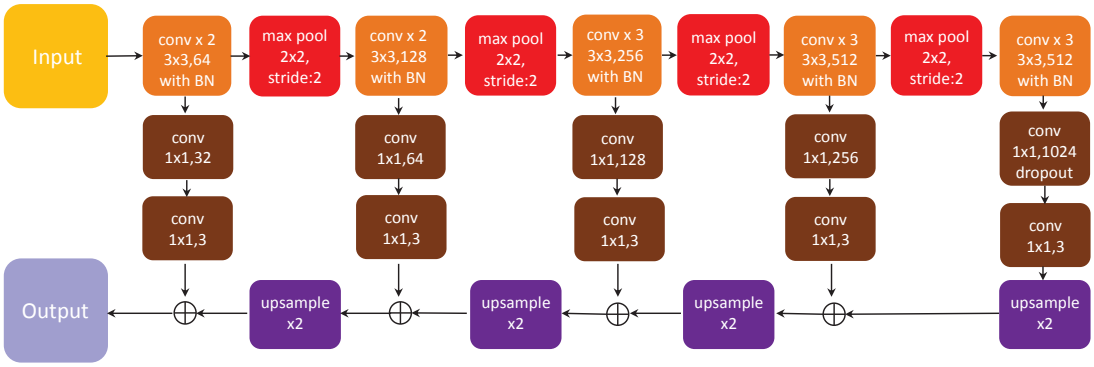


Figure 4.4: Architecture of CNN of the semantic branch. We add batch normalization to the five convolutional layers adapted from FCN. Five up-sampling operations are applied to make the final output the same shape as the input image.

if its appearance descriptor is close to the appearance descriptor of background clusters. In the last stage, the overall tracking result is generated by merging the identified foreground clusters together and being smoothed among temporal and spatial axis.

4.3 Experiment

Implementation Details The convolutional network of the semantic branch has been modified from that of FCN, and the architecture is illustrated in Figure 4.4. We introduce batch normalization after every convolution layer of the network. Also, we employ five upsampling operations to make the final output the same shape as the input image. When training the network, we additionally introduce an auxiliary loss function on the top of the fifth convolution layer to make the training more stable. For the convolutional network used by the optical flow branch, we use the pre-trained network parameters instead of fine-tuning the net on DAVIS [128] . We use the thin models which have $\frac{3}{8}$ of the channels corresponds to FlowNetS and FlowNetC.

The source code of this work will be accessible to on[1]. Please refer to our

[1] `https://github.com/ZijingChen/trackBySeg`

project page to see all the experiment results[1].

Data and Evaluation In this work, we evaluate the proposed tracking method, along with several state-of-the-art trackers on the densely annotated dataset for video trackers [128] (DAVIS dataset). The video contains challenges such as fast motion, shape complexity, and deformation. Besides, the pixel-accurate annotations are ideal for our requirements. Using the DAVIS, we have 30 video clips of training, which include 2079 images. To illustrate the detailed performance of each method on different kinds of tracking conditions, we randomly pick out another 15 video sequences from the remaining set of DAVIS as our evaluation set. The target in our evaluation set can be a single object like a dancing girl. It can also be multiple objects that connected with each other, for example, the *soapbox* video. Since our method is based on bi-channel FCN, we call it FCN^2 tracker.

We refer to the pixel-level ROC curve as the basic evaluation metric. The ROC curve refers to the receiver operating characteristic curve, where true positive rates are plotted against false positive rates at various threshold settings, which correspond to y- and x-axis respectively. In particular, our model gives pixel-by-pixel predictions of class probability, ROC is calculated by varying the classification threshold θ, (i.e. $I_{i,j}$ is predicted as target if $P(I_{i,j} = target) > \theta$. For trackers representing target using the bounding boxes, say, a tracker predicting a box B^*, we generate a series of boxes, centred at the centre of B^*, with varying sizes $\{B_1, B_2, \ldots\}$. ROC curve for the tracker is calculated by predicting the target as pixels within $B_1, B_2,$ respectively.

Our performance is compared with state-of-art trackers: siamese-fc [153], CF2 [106], CSK [70], STRUCK [63], DSST [35], and L1APG [8]. Fig. 4.5 presents the results of the compared trackers in bounding boxes and the proposed method in the probability map. The presented frames come from 6 challenging video sequences which include in-plane rotation, large-scale deformation, ambiguous edge and so on. The illustrated results demonstrate that our method is robust to a various challenging transformation of the target while other trackers become quite vulnerable. For example, when tracking the dancers, many trackers cannot

[1] `https://sites.google.com/site/tbdtracker2017/`

tightly cover the target due to significant deformations. Instead, the proposed method can still predict precise foreground layout for the target.

Figure 4.6 and figure 4.7 show the ROC of our algorithm and compared trackers. Each frame has its own ROC. However, we only report the average value of ROC to present a statistic result. To illustrate the performance of evaluated methods during different periods of the video sequence, we divide each video sequence into ten separate parts according to arrival orders. The results presented in the figure supports that the proposed algorithm achieved superior tracking performance, which is consistent with the intuitive assessment shown in Fig. 4.5. In specific, with tracking through more frames, the ROCs of all trackers deteriorate due to drifting and failures. Nevertheless, FCN^2 tracker remains superior to rival methods.

4.4 Conclusion

This chapter presents a new approach for visual object tracking based on bi-channel FCN that 1) produces a finer tracking result and 2) works for the generic object without fitting the network to the appearance of any specific object class, which needs a large scale of training data. Our model can extract the temporal relationship between two observations of a target that works together with optical flow information to generate a robust tracking result. In future work, we plan to explore extensions that could encode more changes in the semantic information of the tracking target.

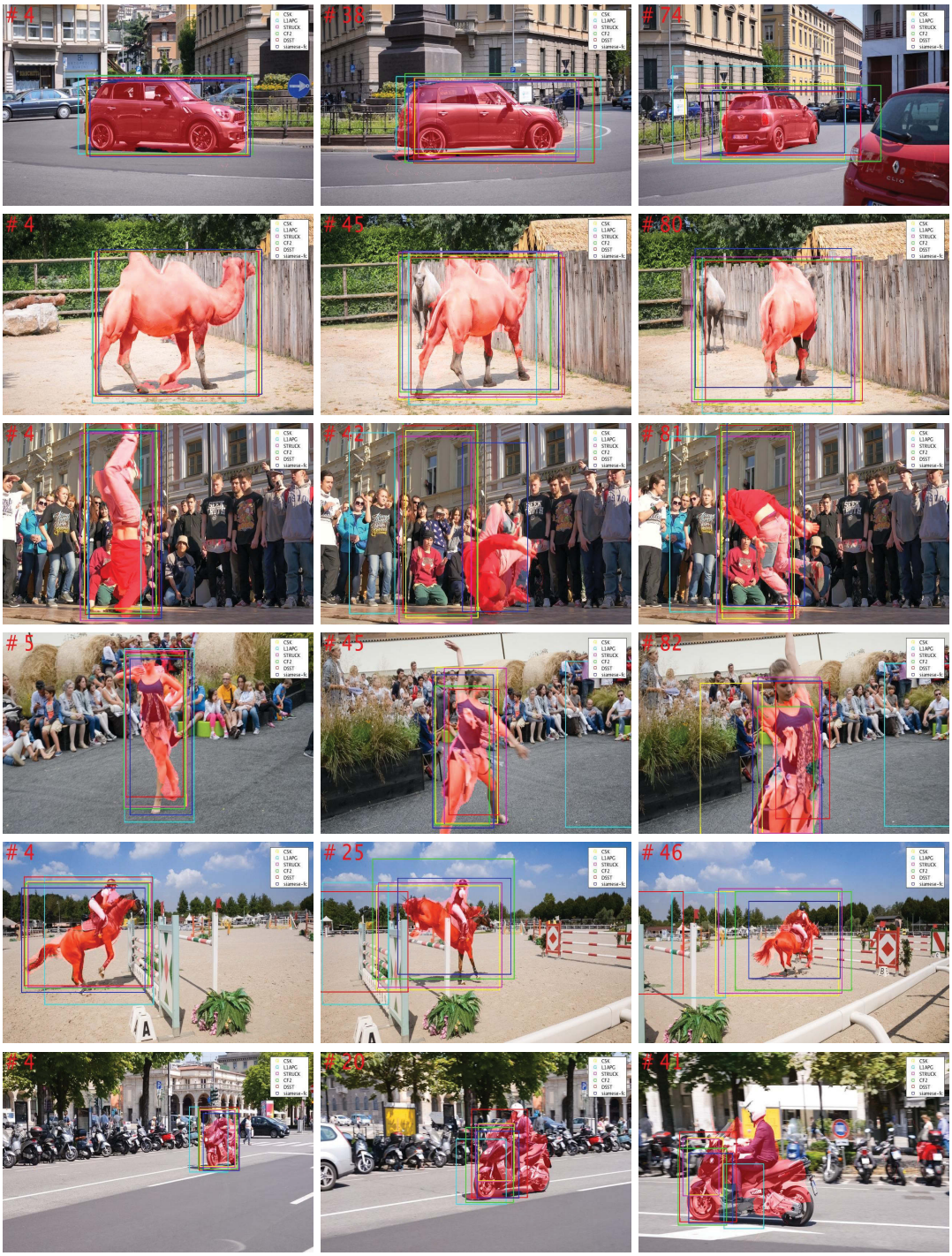


Figure 4.5: Qualitative comparison among trackers. Our output is marked in red shadow. The result of the other trackers are shown by bounding boxes.

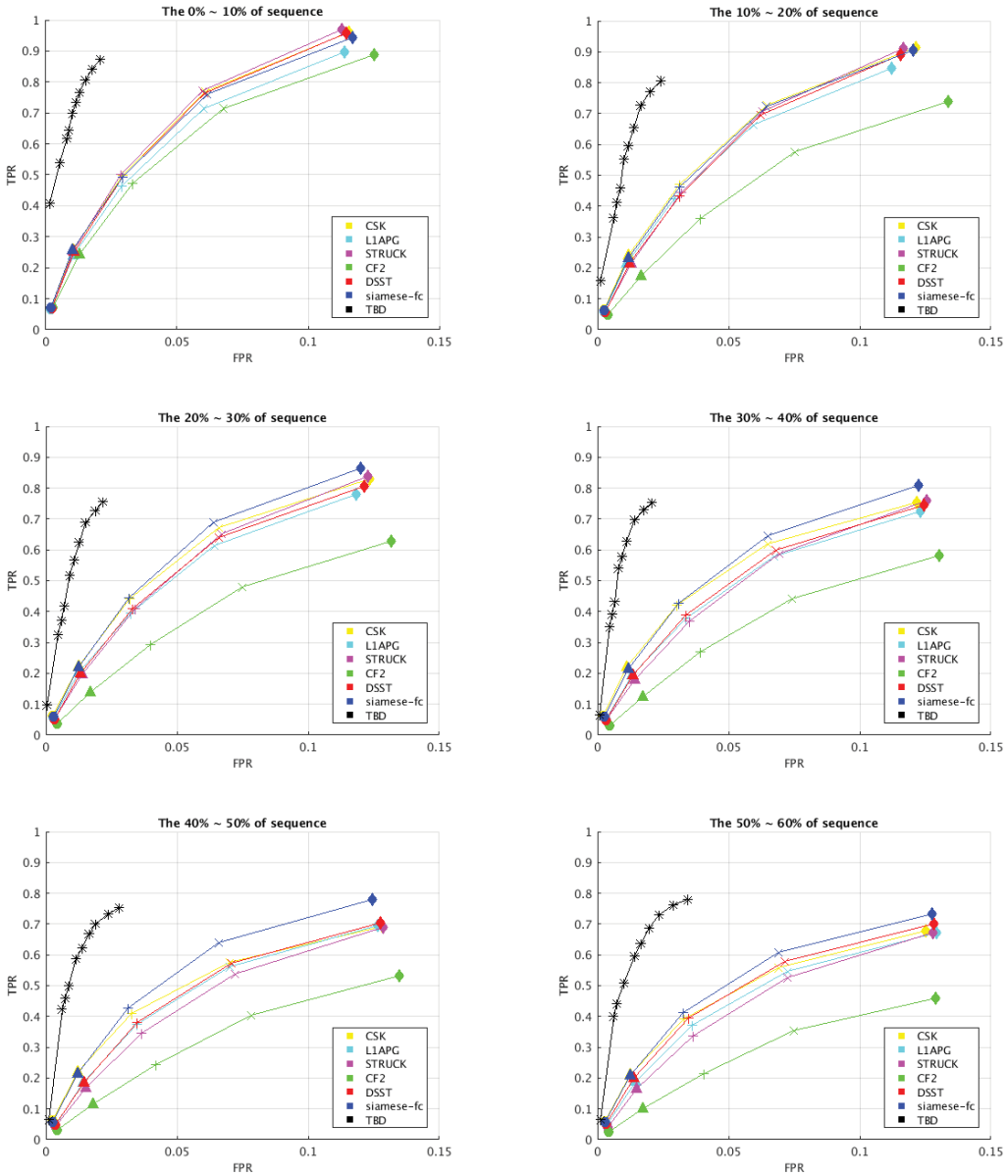


Figure 4.6: ROC, from the beginning to the 60% of a video sequence. Our output is shown by black lines marked with stars. The rest of the other trackers are shown by curves in color.

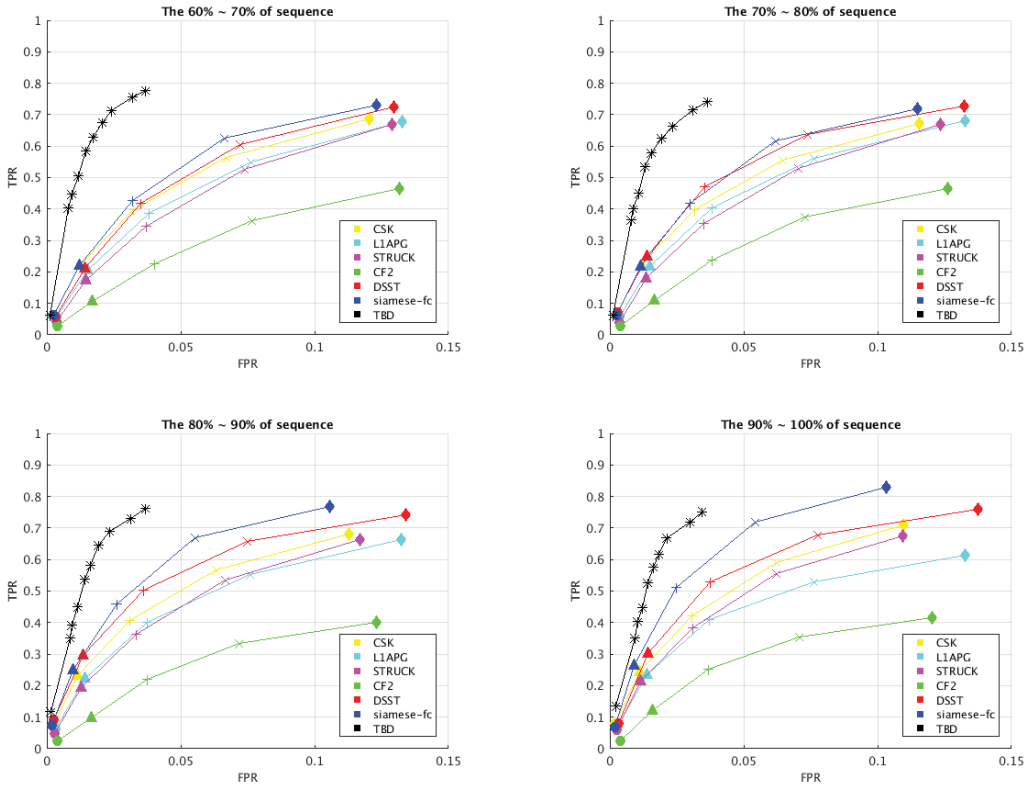


Figure 4.7: ROC, from the 60% to the end of a video sequence. Our output is shown by black lines marked with stars. The rest of the other trackers are shown by curves in color.

Chapter 5

Learn to Focus on Objects for Tracking-by-Detection

The tracking-by-detection strategy is effective in tackling the visual tracking problem. It considers tracking as an online detection problem. In practice, researchers often train a detector on the fly and use it to identify the target from the background image areas. In this chapter, using the tracking-by-detection strategy, we introduce a novel algorithm to help the online detector better focus on the target in the face of challenging issues. Our proposed method substantially improves the robustness and accuracy of a tracker.

5.1 Introduction

In recent years, tracking-by-detection methods have become increasingly popular, achieving excellent performance in the tracking problem. This is because such kinds of tracking methods are more robust against appearance variations and drifting. It is also easier for the tracking-by-detection methods to re-locate objects in a video sequence which have disappeared, especially when tracking objects such as pedestrians and cars.

Despite the advantages, tracking-by-detection methods are extremely time-consuming due to the enormous model complexity brought by the employed detector. Since an object detector needs to train a classifier to identify objects, re-

searchers tend to introduce deep convolutional neural networks (DCNNs) to fulfil the classification tasks considering their outstanding expressive capacity. However, training and using DCNNs could be computationally intensive because they generally have a large number of parameters. As a result, a tracking-by-detection method becomes significantly slow if the DCNN-based detectors attempt to find a target from a large number of image patches. To address this issue, researchers have introduced object proposals, which is a set of windows that may contain the objects, to reduce the computational complexity of a DCNN-based detector.

The utilization of object proposals can accelerate the detection process by identifying and discarding a large number of obvious background patches with the help of object proposal generators. An object proposal generator commonly accepts as input the low-level or convolutional neural network (CNN) features, and outputs a short-listed set of candidate detection windows for the classifier to make the final decision. Compared with sliding window-based detectors which straightforwardly scan over image locations in an exhaustive way to generate the candidate object area, the employment of an object proposal addresses two fundamental limits of the exhaustive search in the sliding window framework: (i) waste of discriminative computations on areas of an image where the presence of *any* object is unlikely, and (ii) waste of training data by *ad hoc* sub-sampling tricks to get the positive and negative samples balanced. Thus, the object proposal has been widely used in state-of-art detection algorithms and largely boosts the performance of the detector, including detectors based on traditional features and classifiers, and those based on the convolutional neural networks [54]. Regarding the contribution of object proposal for object tracking [67], it indicates the possible location and shape of the target in a newly arriving frame and relieves the computational burden.

However, the generation of object proposals remains challenging. The accuracy of proposals is usually reduced when dealing with complex vision scenarios, where the objects can appear at arbitrary image locations, of different scales, within different categories, and their number may vary across different images. This is due to the fact that the scheme is of the pre-discriminative stage, i.e., the algorithm is NOT allowed to access any object category-specific information, and yet must provide *full coverage*: guarantees that any object to be detected is

covered by one of the candidate windows. Thus for traditional methods relying on global or local image features, the focus of object proposal research has been duly put on inventing various measurements that gauge how likely an area may contain any interesting target (known as objectness measure), and investigating the low-level image cues based on which the measurements and proposals are constructed [2, 132, 183, 200]. These object proposal generators based on hand-crafted features need to process tens of thousands, if not hundreds of thousands of candidate windows to generate object proposals which could ensure full coverage of potential objects in an image. Besides, since the features used to judge the quality of the proposal may not be so powerful, proposals contain background areas can be mistakenly selected as good references for the object detector and spoil the performance of the detector. Additionally, with more powerful features, the deep learning based methods employ the Regional Proposal Network (RPN) to simultaneously predict object bounds and objectness scores at each position as a reference for the object detector. With RPN, the number of object proposals has been reduced, and the overall quality of the proposals remains at a high standard. Thus RPN or similar modules has become a needful part of deep learning based detectors, such as [133] and [102]. One of the main limitations of RPN is that it only uses small convolution kernels, i.e. 3x3, to propose bounding boxes for all the objects presented in the images. In deep learning, the small convolution kernel means that the corresponding receptive field is also small. In practice, a small receptive field will force the network to make a judgement about whether there exists an object based on a limited range of visual features. Without sufficient visual features, it will become difficult for RPN to propose bounding boxes for objects of different sizes and shapes robustly and accurately. According to the above, we propose a complementary method to work with object proposals, which can be used to improve the performance of both the traditional detectors and deep learning based detectors. It also avoids above limits and become more flexible and accurate.

Our works are motivated by biological patterns: Humans focus on the target in the view fastly. In fact, humans and many animals do not look at a scene in fixed steadiness; instead, the eyes move around, locating interesting parts of the scene. The paper published in Nature [114] reveals human observer can move their

eyes toward the target with adaption. The recent research in the deep learning book [57] indicates that human glimpses the most visually salient or task-relevant parts of a scene, which is a small area rather than the entire scene. Based on these, we can find that human can efficiently and accurately focus on the target because they can adapt their focus by utilizing the context information.

Considering above, our work presents an alternative development of the object proposal algorithms to improve the performance of tracker called TRM Tracker (TRMT). The main idea is to utilize the context information around possible object areas to help the tracker better focus on the target and output more accurate tracking result. The image cues around the existing proposal act as context to redeem the partially aligned windows in any set of candidate windows and thus improve the overall proposal. The proposed scheme is called TRansformation Model (TRM) which acts as a compliment and is orthogonal to the efforts of sophisticated searching schemes. It can be applied flexibly to the proposals based on hand-crafted features, or proposals produced by CNN features. For the former, a learning-based TRM is developed to iteratively address the translation and deformation bias from misalignment. In particular, the transformation model (i) moves a candidate window on the image plane to where a nearby object may present, and (ii) adjusts the dimensions of the window to achieve more accurate coverage of the object. For the latter, the TRM is implemented by a Focus Proposal Net (FoPN) which is constructed of several convolutional layers. It first generates candidate proposals similarly to RPN. Then, the context information around the candidate area is extracted by specifically designed dilated convolution kernels. After that, the location and shape of the candidate area are adjusted iteratively to help the refined proposal focus on the target. Experiments on real-life images show that the transformation model improves existing proposals. In addition, the performance of the detector and tracker can be improved when loaded with the proposed transformation model. In details, we have observed a statistically significant increase in the coverage of the objects in the images from a given set of candidate windows to one that is transformed by our model. Besides, the overall performance of the detector is improved when providing FoPN. Finally, the overlap rate on the test tracking sequence is increased, which indicates that a more robust tracker can be obtained. With our proposed method, the quality of every

object proposal can be improved and thus the performance of object detection algorithms, as well as the tracking-by-detection algorithms, would be significantly advanced even when using a small set of proposals. Empirical studies show that our method boosts the detection and tracking performance with a great margin, demonstrating the effectiveness of the proposed components.

This chapter is organized as follows. In Section 2, we summarize the works most related to ours. Then we describe the proposed scheme in Section 3, including the application of the proposals generated by hand-crafted features or by convolutional neural networks. We present experiments in Section 4 and finally draw conclusions in Section 5.

5.2 The Transformation Model

This transformation model is developed to serve the proposal-based object detectors. The main idea is to redeem the partially aligned windows to improve any existing proposal. In particular, given an image $\mathcal{I}$, suppose $\mathbf{B}$ includes all the rectangular areas of interest. Then the object proposals are generated by the function: $\phi : \mathcal{I} \rightarrow \mathbf{P}_{\mathcal{I}} \subset \mathbf{B}$, where $\mathbf{P}_{\mathcal{I}}$, the output of this function, is a set of candidate windows. The windows of $\mathbf{P}_{\mathcal{I}}$ are expected to contain all potentially interesting objects in $\mathcal{I}$. Instead of designing yet-another ϕ, what this transformation model contributes is to construct learning based statistical models for adjusting windows: $\psi : \mathcal{I} \rightarrow b'|b$, where $b, b' \in \mathbf{B}$. Compared with b, b' is supposed to be more focused on the object. By analyzing the context information, the model ψ adjusts proposals in $\mathbf{P}_{\mathcal{I}}$ so that they are transformed to areas more likely to contain objects. Thus $\phi \circ \psi$, which generates proposal and then adjusts them, becomes a better proposal scheme. With such structure, the model can work with any existing proposal scheme (such as [157]). In practice, the proposed scheme can be implemented with hand-crafted features to improve the traditional object proposal algorithms such as [157, 200], and with CNN features that can be injected into existing object detection framework easily. Before we introduce the details on how to build the transformation models ψ to predict appropriate adjustments for different windows, it is helpful to discuss some aspects that may be of concern: (i) our focus is to alter the proposal framework; flexibility should

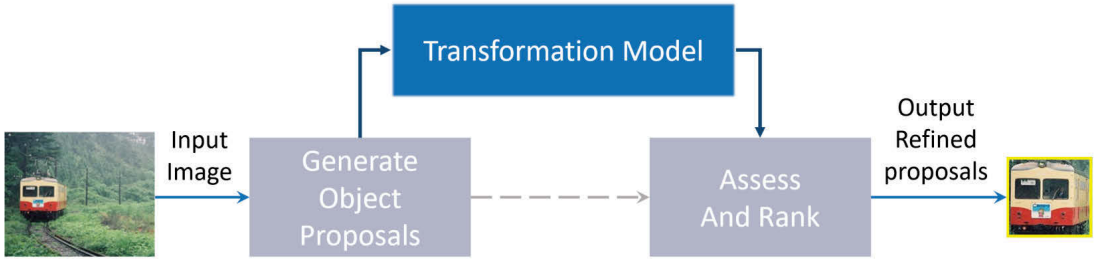


Figure 5.1: Focus on objects with transformation models. The transformation model transforms partially aligned original proposals and focus them on objects.

be given to the particular implementation of the ψ model. Particularly, when representing the image context, any feature used by existing schemes ϕ could be reused for building ψ, so that the extra steps would incur negligible computational overheads. (ii)If coverage of the object of interest is to be guaranteed in a certain stochastic sense by the proposal system, then the introduction of ψ can be considered as making the system more tolerant to the original proposals ϕ. As an example, if the original system needs approximately 10^5 windows like in [2], we can afford a ψ model to achieve similar performance by only using 500 windows. (iii) Since ψ examines local image context to adjust windows, some partially aligned windows are required to work with. Based on the above concerns, as Fig. 5.1 shows, the inputs of the designed transformation model are original proposals and context represented by image features taken within and around the proposal of question, while the output is the instruction on how to adjust the proposal window to better cover the target.

Suppose the initial state of a proposal is marked by the top-left corner location, the width, and the height of the proposal. Then this state can be marked by a 4-dim vector $\mathbf{b} = [xmin, ymin, width, height]$. Suppose the context information around this propsoal $\mathbf{b}$ is describe by a funciton $g(\mathbf{b})$, the transformation model $\psi(\cdot)$ analyzes $g(\mathbf{b})$ and iteratively adapts $\mathbf{b}$ to the target's location and dimension.

The function of the transformation model is described as:

$$\mathbf{b_{k+1}} = \psi(\mathbf{b_k}) \tag{5.1}$$

$$\psi(\mathbf{b_k}) = \mathbf{b_k} + \delta(\mathbf{b_k}) \tag{5.2}$$

$$\delta(\mathbf{b}) = \mathrm{h}(\mathbf{W}, g(\mathbf{b})) \tag{5.3}$$

The function h$(\cdot)$ in Eq. 3 takes the $\mathbf{W}$, the weights learned with training data, as well as the context information as input, and outputs the change in location and shape of the proposal. Then, combining the change, $\delta(\mathbf{b})$, with the state of the existing proposal, a new proposal can be obtained as Eq. 2 shows. The subscript k in Eq. 2 indicates the k-th iteration.

Next, we will introduce the details of designing the transformation model (TRM), which is different for the proposal generation that based on hand-crafted features or CNN based features.

5.2.1 TRM with hand-crafted features

For refining proposals generated by hand-crafted features, two types of transformation are considered in our work. One of them translates the centre of a candidate window within a range to a different position that may be closer to the object and the other adjusts the dimensions of a window to make it cover the object more properly. Correspondingly, the TRM is composed of two sub-models, the translation model and the deformation model. The working flow and the output of each model are as Fig. 5.2 shows. In this case, $\delta(\mathbf{b_k})$ is related to the location change in the vertical direction (d_v), the horizontal direction (d_h), and the dimension change in width (Δw) and height (Δh). The translation model outputs the location adjusted in d_v and d_h while the deformation model handles the shape change in Δw and Δh.

5.2.1.1 Translation model

By analyzing context information, the translation model moves the original proposals that are partially aligned with interesting objects to achieve better alignment. Since any offset of a window can be decomposed into vertical and horizontal

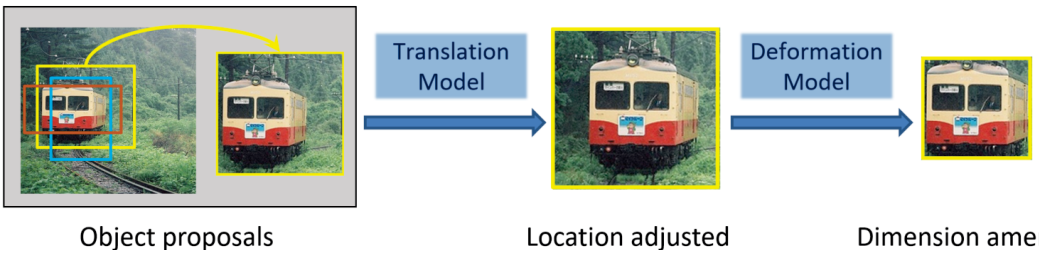


Figure 5.2: The working flow of improving proposals based on hand-crafted features. Original proposals have been generated on the input image. With the processing of the translation model, the locations of these proposals are adjusted to better align the target. Then with the deformation model, the scale and shape of the proposal are amended to better focus on the object.

components, the desired outputs of this model are two rectification movements along corresponding directions. In order to facilitate the framework to work with proposals of variant sizes, its output is rescaled to $[-1, 1]$ as proportions to the scale of the original window. The sign of the output denotes the direction of movement, where the negative sign refers to the movement in opposite direction compared to the positive sign. Suppose τ_v and τ_h are outputs of the translation model along vertical and horizontal direction, a pair of regression models can be constructed based on image features:

$$\tau_v = \hat{h}(\langle \mathbf{W}_v, \mathbf{f}_v \rangle) \tag{5.4}$$

$$\tau_h = \hat{h}(\langle \mathbf{W}_h, \mathbf{f}_h \rangle) \tag{5.5}$$

where $\mathbf{f}_v$ and $\mathbf{f}_h$ are vertical and horizontal feature vectors extracted from the image, which are related to contexts in vertical and horizontal directions (will be explained later). $\mathbf{W}_v$ and $\mathbf{W}_h$ are the corresponding learned weight vectors. $\hat{h}(\cdot)$ can be implemented with squashing function and we choose $tanh(\cdot)$ to be the function. Afterwards, the desired transformation movements, which are proportional to the height and width of a window, are calculated by following equations:

$$d_v = \tau_v \cdot H_{\mathbf{b}} \tag{5.6}$$

$$d_h = \tau_h \cdot W_{\mathbf{b}} \tag{5.7}$$

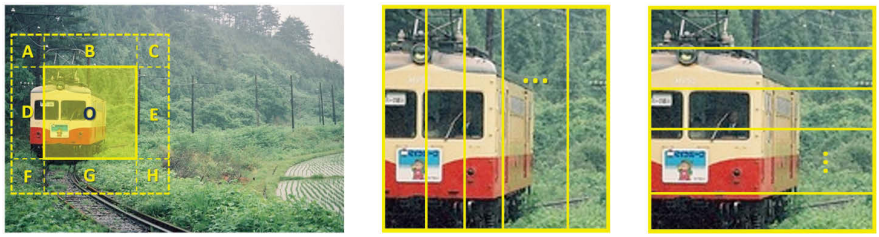


Figure 5.3: Area arrangement of feature extraction for one proposal. Above figures show how the characters of the context around and within an original proposal (represented by the shadowed area marked as 'O' in (a)) are extracted. As in (**a**), surrounding areas are organized in a grid. These grid cells can be combined to generate surrounding features. For example, if we use braces to represent a combination of cell areas, then features can be extracted from {A, B, C, D, O, E} and {F, G, H} separately. In addition, the proposal area, O, is partitioned in two passes, one horizontally and the other vertically, generating internal horizontal (**b**) and vertical (**c**) features.

where $H_{\mathbf{b}}$ and $W_{\mathbf{b}}$ are height and width of the proposal $\mathbf{b}$.

For solving Eq. 5.4 $\sim$ Eq. 5.7, we first introduce how to build the features $\mathbf{f}_v$ and $\mathbf{f}_h$, then discuss the training of weight vectors,$\mathbf{W}_v$ and $\mathbf{W}_h$.

To represent the spacial characteristics of the context efficiently, the arrangement of the feature extraction areas is carefully designed. Since the candidate proposal may only partially overlap with the potential object, when constructing $\mathbf{f}_v$ and $\mathbf{f}_h$, image features both around and within the window areas are considered to tackle various cases. Surrounding features are extracted to judge the offset of a proposal window when the object is located in the vicinity of it or crossing it. As Fig. 5.3(a) shows, the areas around the original proposal (marked by O) are organized by a grid, thus vertical and horizontal surrounding feature vectors $f_{v\cdot surr}$ and $f_{h\cdot surr}$ can be extracted from areas represented by different combinations of the grid cells (marked by A to H) and the proposal O. For example, suppose we use braces to represent a combination of cell areas. Then by dividing the area into an upper part and a bottom part, the features extracted from {A, B, C, D, O, E} and {F, G, H} can form two elements in $f_{v\cdot surr}$. Similarly, features extracted from {A, B, D, O, F, G} and {C, E, H} form two elements in $f_{h\cdot surr}$ respectively. On the other hand, internal features are helpful to capture the presence of the

object when it is partially or fully contained within the candidate window. As Fig. 5.3(b)(c) illustrates, the internal area of the window 'O' is scanned twice by the feature extractor, once vertically and thence horizontally, to produce the corresponding feature vectors $f_{h\cdot in}$, $f_{v\cdot in}$. These two-direction internal partitions serve well for predicting the vertical and horizontal offsets to move the current candidate window for a better alignment. The grid partition in Fig. 5.3(a) is not used to produce internal features for the following reasons: (i) features can be extracted more efficiently, which saves computation power; (ii) the required training sample size can be reduced; (iii) a strip area, which is larger than a cell area, contains richer information for the feature to catch.

The features extracted in each surrounding or inner area are represented as a vector. Also, the translation model is open to any type of image features as long as it is descriptive and efficient to compute. In this work, we use the saliency descriptor as in [2] because this feature is widely used in existing object proposal algorithms. Besides, with the help of integral saliency image, which is a quick and effective way of calculating the sum of values in a rectangular area within the image, the feature computation will not incur any overhead if our adjustment is used in conjunction with these proposal algorithms.

Lastly, one vertical feature vector and one horizontal feature vector for each candidate window are generated by concatenating feature vectors belong to the corresponding directions as follows:

$$\mathbf{f}_v = [f_{v\cdot in}, f_{v\cdot surr}] \tag{5.8}$$

$$\mathbf{f}_h = [f_{h\cdot in}, f_{h\cdot surr}] \tag{5.9}$$

In our implementation, both $\mathbf{f}_v$ and $\mathbf{f}_h$ are 22 dimensional real-valued vectors. Each element of them is calculated with several simple addition and subtraction operations on the pre-calculated integral image which is generated from the input image.

Compared with powerful learning models such as decision tree and structured SVM, a regression model is chosen for training the weights $\mathbf{W}_v$ and $\mathbf{W}_h$. This is because this thesis aims at highlighting the effectiveness of the active learning scheme rather than the contribution of learning model itself. Furthermore, all

features can be significant in the logistic regression model after analyzing the significance of features in different models statistically.

In our experiment, the nonlinear least squares regression is used. As a result, $\mathbf{W}_v$ and $\mathbf{W}_h$ are respectively trained by optimizing the following problems .

$$\min_{\mathbf{W}_v} \sum_i \left(\hat{\tau}_{vi} - \hat{h}(\langle \mathbf{W}_v, \mathbf{f}_v \rangle) \right)^2 \tag{5.10}$$

$$\min_{\mathbf{W}_h} \sum_i \left(\hat{\tau}_{hi} - \hat{h}(\langle \mathbf{W}_h, \mathbf{f}_h \rangle \right)^2 \tag{5.11}$$

where $\hat{\tau}_{vi}$, $\hat{\tau}_{hi}$ are labeled offset ratios of the i-th training sample and the squashing function $\hat{h}(\cdot) = tanh(\cdot)$.

5.2.1.2 Deformation model

The dimensions of proposals are adjusted by the deformation model. Since the deformation model addresses the uncertainty about the scale and shape of the object, the searching space in the detection process can be inflated. The translation model and the deformation model can work jointly and significantly to improve the accuracy and coverage of object proposals.

Given a proposal, the deformation model enhances the chance that the proposal covers an object accurately by iteratively attempts to stretch or shrink it. The deformation process alternates between the horizontal and the vertical directions. In each iteration, the proposal is shrunken or stretched according to the context around it. Specificially, the deformation model updates the latest shape and scale of the proposal in each iteration with the variance in width (Δ_w) and height (Δ_h). The iteration will be ended if a local optimal dimension of the proposal has been reached, according a *deformation evaluation model*:

$$C(\mathbf{b}) = \Omega(g(\mathbf{b})) \tag{5.12}$$

$$= \alpha_1 \cdot \log(S(\mathbf{b})) - \alpha_2 \cdot \log(A(\mathbf{b})) \tag{5.13}$$

where $C(\cdot)$ estimates the compactness of the candidate window $\mathbf{b}$, which is depended on the context represented by $g(\mathbf{b})$, and Ω is the scoring function. $S(\mathbf{b})$ is the sum of saliency measures within $\mathbf{b}$, $A(\mathbf{b})$ is the area of the window. $S(\mathbf{b})$

and $A(\mathbf{b})$ jointly defined $g(\mathbf{b})$. α_1 and α_2 are balancing weights between $S(\mathbf{b})$ and $A(\mathbf{b})$. As [2] mentions, a distinct character of the object is "stands out as salient". Thus $S(\mathbf{b})$, which can be efficiently calculated with a pre-calculated integral image, is used to capture holistic characters in the proposal and should be positive correlation with $C(\cdot)$. Besides, $A(\mathbf{b})$ is adopted not only because of the significance of $S(\mathbf{b})$ is related to the area, but also for punishing the windows being excessively large. Thus $A(\mathbf{b})$ is considered as being negatively correlated to $C(\cdot)$. It is worth noting that unlike the intersection-over-union (IoU) score which measures the degree of the overlap with the ground truth, $C(\mathbf{b})$ is computable on both the training and the test images, i.e., evaluation of Eq. 5.12 does NOT rely on any annotation of the image. In summary, the iteratively change in the aspect ratio of the proposal can be represened by:

$$\Delta w_k, \Delta h_k = \arg\max_{\Delta w, \Delta h} \{C(\mathbf{b_{k-1}} + [0, 0, \Delta w, \Delta h]) - C(\mathbf{b_{k-1}})\} \tag{5.14}$$

The details of the algorithm are explained in Algorithm 2. The parameter λ referred in lines 6,10 and 22 is introduced to resist interference. The parameters α_1, α_2 and λ are chosen so that $C(\mathbf{b})$ reflects the true degree of overlap between $\mathbf{b}$ and some ground truth annotations. In our empirically study, letting α_1 and α_2 be 1.0 led to effective evaluation criteria, and λ is learned from the training set so that Algorithm 2 produced satisfactory deformed proposals.

5.2.2 TRM with CNN features

Besides hand-crafted visual features, our proposed transformation model can also benefit the proposal generators that refer to CNN features as a more powerful representation of objects. Following the basic concept of improving quality of proposals by using context and adaptation operations, it is easy to implement the proposed method by directly injecting appropriate convolution operations in the existing CNN-based proposal generation pipeline and thus make the whole processing flow have an end-to-end architecture. We consider the convolution blocks that introduced to acquire the value of $\psi(\mathbf{b_k})$ as Focus Proposal Net (FoPN).

In FoPN, we attempt to achieve focusing procedure by actively including rich

context information in diversified spatial ranges and by introducing a cascaded structure to fulfill the task of iterative proposal adapting procedure.

5.2.2.1 Context Module

The context information is crucial for adjusting the proposal. Instead of transforming surrounding hand-crafted features into the expected context feature vectors as mentioned in the previous section, we attempt to incorporate the contexts by using different shape-transformed kernels in convolution operations. Furthermore, the contexts in different ranges are also considered with the help of dilated convolution kernels [19].

In specific, as Fig. 5.4 shows, three kinds of shape-transformed kernels are utilized to capture different kinds of contextual features. First, to better understand object structure in a vertical view, we design a bar-shaped (Fig. 5.4(a)) convolution kernel to extract vertical contexts. To explore the context information in multiple spatial ranges, we use several dilated convolutions to implement the convolution kernels which applied to the original spatial area, and to wider spatial ranges. The dilation rates d are set to $d = 1$, $d = 2$ and $d = 4$. The horizontal context is extracted in a similar way, but the kernels are applied in a horizontal direction (Fig. 5.4(b)). We believe that these bar-shaped convolution kernels are essential for adjusting the location of the originally proposed areas because features of a single direction (vertical or horizontal) can be concentrated. In addition to the bar kernels, there is a third kernel, which is used to explore the pattern of the surrounding areas of the candidate as Fig. 5.4(c) shows. Contexts lie in rectangular areas can be captured to facilitate judging the shape of interested objects.

5.2.2.2 Iterative proposal adapting

In addition to using context to adjust proposal states, the adjusted proposals can be further refined to have improved coverage about objects. Similar to humans, we adopt the use of iterative adaptation procedure to refine proposals step by step. However, the difficulty of employing iterative processing module for CNNs is that existing neural networks are generally defined by directed acyclic graphs,

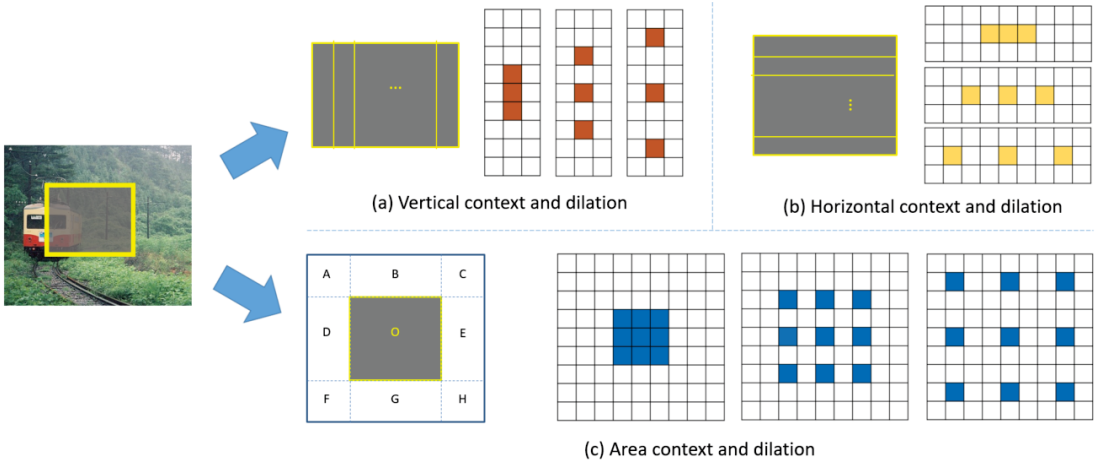


Figure 5.4: Explore the context in multiple spatial ranges. Three kinds of kernels are utilized here. The bar-shaped convolution kernels in (a) and (b) are designed to extract vertical patterns and horizontal patterns separately. The area kernel in (c) is used to explore the pattern of surrounding areas. Besides, to analyze the context information in multiple spatial ranges, the Atrous convolution is applied on these kernels.

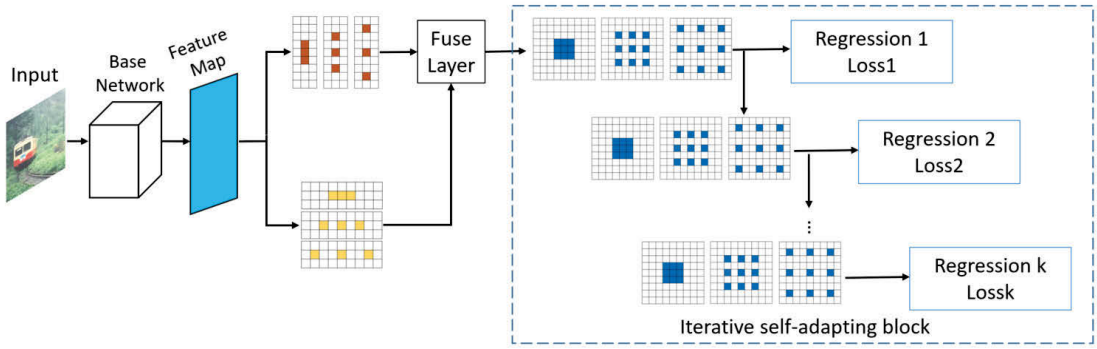


Figure 5.5: Focus proposal net. With the iterative self-adapting block, the transformation parameters predicted at step k will be combined with parameters obtained at step $k-1$ which helps the proposal to gradually focus on the target.

meaning that we are not able to directly use processing loops to implement iterative adaptation. As a result, we propose to achieve the iterative operations by employing the Iterative Self-adapting block which has a cascaded structure as illustrated in Fig. 5.5.

In specific, we predict the transformation parameters in each processing step of the cascade. The transformation parameters predicted at step k will be com-

bined with parameters obtained at step $k-1$. Following this pattern, the final transformation parameters can be predicted by fusing the outputs of all the previous processing steps in the cascade in an iterative manner. In this section, we make the network produce the transformation parameters for both location and shape adjustment. Suppose we have context feature representation, $g(\boldsymbol{b_k})$, for bounding box $\mathbf{b_k}$ at k-th step. Then the proposed FoPN learns to predict $\delta(\boldsymbol{b_k})$ directly from $g(\boldsymbol{b_k})$. SmoothL1 [133] is used to optimize the weight parameters for the mapping from $g(\boldsymbol{b_k})$ to $\delta(\boldsymbol{b_k})$. According to Eq. 5.2, the fusing operation on results from different steps is implemented by the addition operation. It is worth noting that we can re-use the kernels that consider surrounding context to produce transformation parameters in the cascade.

5.2.2.3 Object detector

The FoPN can be easily embedded into prevailing the detection framework to form an advanced version of the detector. The working flow of the advanced object detector is as Fig. 5.6 shows. It takes gray or colour image as the input, and uses a base network to generate feature maps which preserved key information of the target object. Then the proposed FoPN processes with the feature map, and generate refined proposes. These proposals are sent to the detection part (e.g., Fast R-CNN), and output the detection result which contains a label marked the category of the target and a bounding box specified the location and shape of the target.

5.3 Experiments

In this section, we empirically evaluate the proposed TRM, as well as the proposal generators, detector and tracker which integrate our transformation models. The experiments are conducted on proposals generated by hand-crafted features and CNN based features. We first examine how the transformation model affects the distribution of the object proposals. Then we show that the effects of our approach on object proposals of individual images can generalize to better object proposals on a large real-life image data set, the Pascal VOC 2007

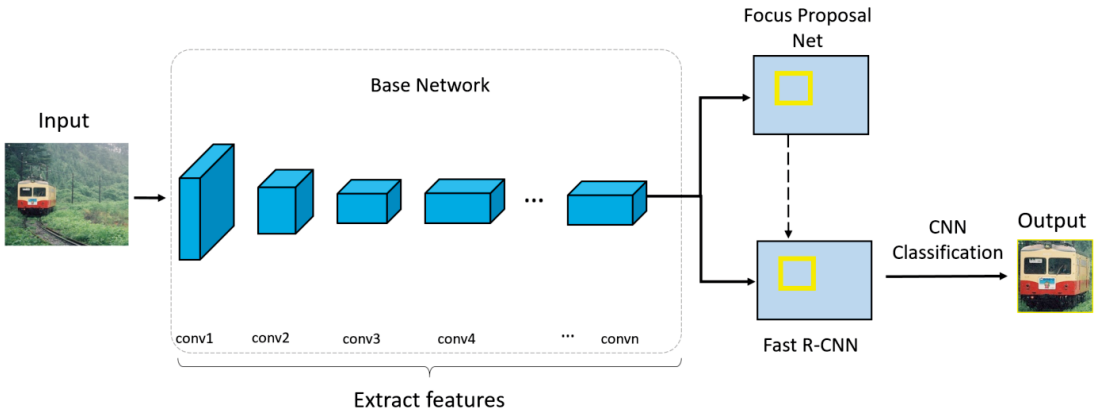


Figure 5.6: The working flow of object detector with FoPN. The detector takes an image as the input and uses a base network to generate feature maps which preserved key information of the target object. Then the proposed FoPN processes with the feature map, and generate refined proposals. These proposals are sent to the detection part (Fast R-CNN), and output the detection result.

set [46]. Besides, the improvement in the performance of a state-of-art object detector, with the embedded transformation model, has also been proved by the results. Finally, the boosted performances of visual tracker that equipped with the proposed transformation model are presented and analysed in this part.

The source code of this work will be accessible to on[1]. Please refer to our project page to see all the experiment results[2].

5.3.1 Compared With Traditional Proposal Methods

For constructing the TRM, the training samples and labels are generated from the training set of the VOC 2007 dataset. Since the ground truth objects have been specified with bounding boxes on each of the training images, our training samples can be obtained by simply adding random location displacements and dimension changes to these boxes. Then, the offsets regarding the ground truth are calculated to generate the training labels. The additional running time of using TRM is depended on the number of candidate object proposals of each frame. In our experiment conducted on the VOC 2007 test set, the average

[1] https://github.com/ZijingChen/focuson/
[2] https://sites.google.com/site/focusproposal2017/

processing time of our model for one proposal is $1.1ms$ on 2.3GHZ E5-2650 CPU without any parallel running tricks.

To judge the improvement of the quality of object proposals based on hand crafted features, we utilize the overlap score taken from [47] to measure the area of the intersection between a proposal $\mathbf{p}$ and a ground truth annotation $\mathbf{g}$ divided by its union:

$$Overlap(\mathbf{p}, \mathbf{g}) = \frac{A(\mathbf{p} \bigcap \mathbf{g})}{A(\mathbf{p} \bigcup \mathbf{g})} \tag{5.15}$$

5.3.1.1 Improvement on proposal population

First of all, how the TRM works on a population of object proposals regarding a variety of real-life images is discussed in details. The beneficial effects are demonstrated by both visual inspection and will be later verified by the statistical investigation.

The effects of the translation and deformation models on real-life images are illustrated in Fig. 5.7. For each test image, six sub-figures arranged in three columns are illustrated and compared to show the performance. From left to right, the three columns correspond to the performance of three stages. They are random boxes as initial proposals, adjusted proposals after applying the transformation model, and further refined proposals with the help of the deformation model. The real-life images in the first row show the 5 top-ranked proposals out of all proposals in each stage. The ranking is according to the scores computed as in [2]. Take the sheep image on the top-left as an example. Due to the small set of proposals (tens as opposed to $\sim 10^5$ commonly used in object proposal schemes), the five selected proposals are only roughly related to the subject. In addition, the proposals at the second stage are now better aligned with the subject than the original 5. Finally, the proposals adjusted by the deformation model at the third stage are clearly improved when compared to the performance of the counter part in the initial stage. The three sub-figures in the second row show the degree of coverage by the entire set of proposals. In these images, the more frequently a pixel is covered by proposals, the brighter red it is. Obviously, the red area in the second stage is dispersive while the red area in the third stage is centralized. This phenomenon proves that, with translation and deformation progress, most

of the proposals can move toward objects and achieve better alignments. TRM also works well with images that contain multiple objects. The performance can be illustrated by the image depicting two dogs running in the snow, or the image where a man is petting a dog. However, there are also situations that our model may be failed as the Fig. 5.8 shows. The man driving the car is missed because his size is too small compared to the car; two cats are identified as one object because one of them is severely occluded by the other.

5.3.1.2 Improvement on data set

To further validate the efficacy of our model on diversified images, we present several quantitative experiments based on the popular Pascal VOC benchmark. We mainly use the VOC 2007 test set for evaluation.

As mentioned previously, our framework should be able to improve any existing proposal generators with substantial boost for the quality of the proposals. To demonstrate this flexibility, we first apply our method to randomly generated proposals. If our method works well with random proposals, it can also improve other proposal generation methods because the generated proposals share the same form (i.e. coordinates for the top-left and bottom-right corner). More specifically, we generate random proposals on each image in the test set and then perform TRM on them. As a result, the horizontal and vertical translation and the deformation are subsequently performed on the proposals. The overlap scores between ground truth annotations and the proposals are calculated with Eq. 5.15, and the accumulative numbers of proposals whose overlaps lie in different ranges are presented in Fig. 5.9. The random proposals that only cover background areas are ignored in this sector because our model targets on partially aligned proposals. According to the statistical results presented in the figure, it can be found that the translation model and the deformation model reduce the number of proposals whose overlaps with the ground truth are lower than 0.2 and enlarge the number of proposals whose overlaps are higher than 0.3. This proves that our model can promisingly improve the alignment of original proposals.

Next, we present several experiments which illustrate how our transformation models improves the performance of proposals provided by popular proposal

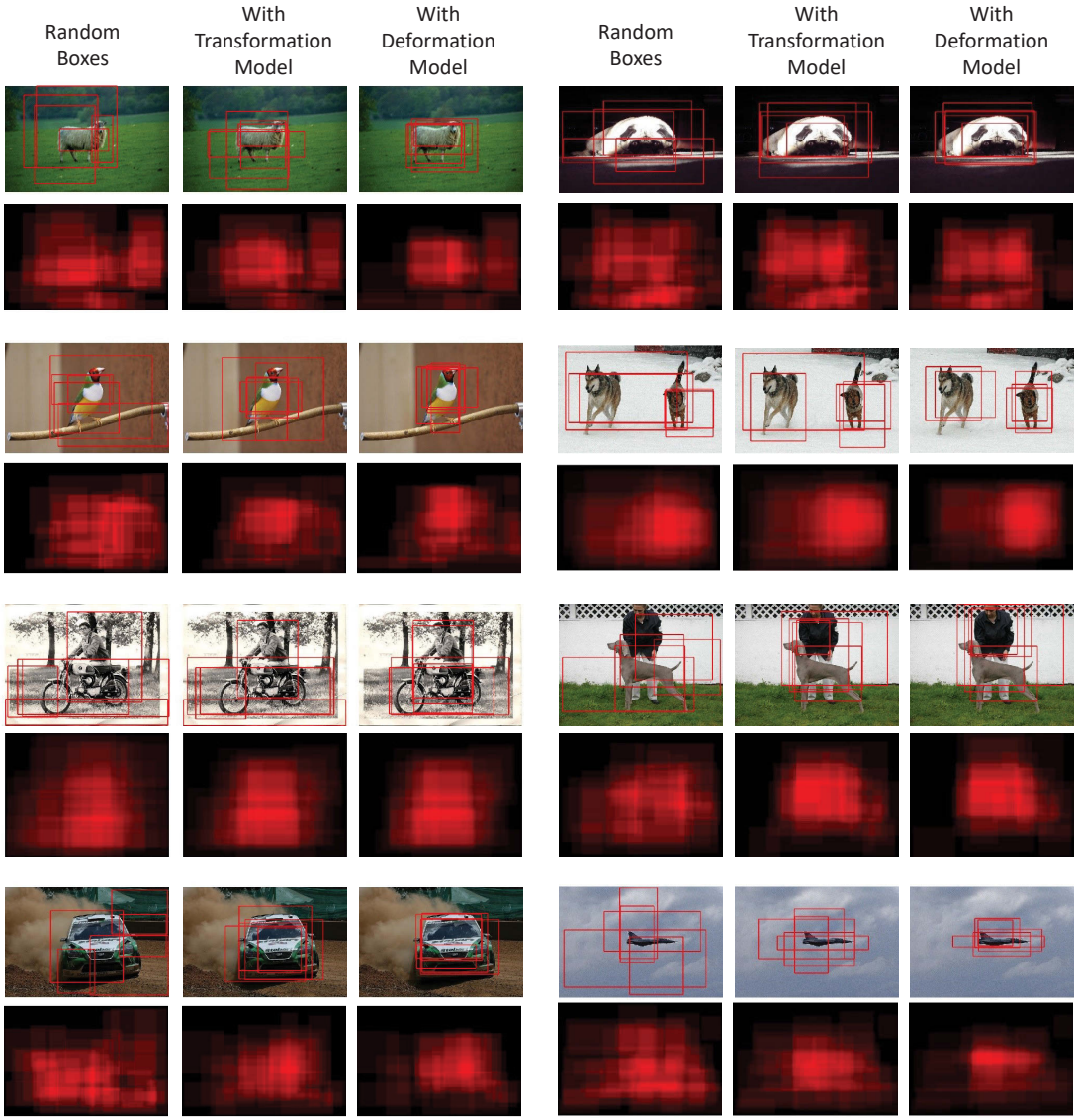


Figure 5.7: Effect of TRM on object coverage by object proposals: visual assessments. Each test image is illustrated with six sub-figures that arranged in three columns and corresponds to 3 stages (from left to right) separately. They are random boxes as initial proposals, adjusted proposals after applying the transformation model, and further refined proposals with the deformation model. The real-life images in the first row show the 5 top-ranked proposals out of all proposals in each stage. The second row shows the degree of coverage by the entire set of proposals in the corresponding stage, where brighter colours are for higher levels of coverage, i.e. when a pixel is included in more windows within the proposal set, its colour will be brighter.

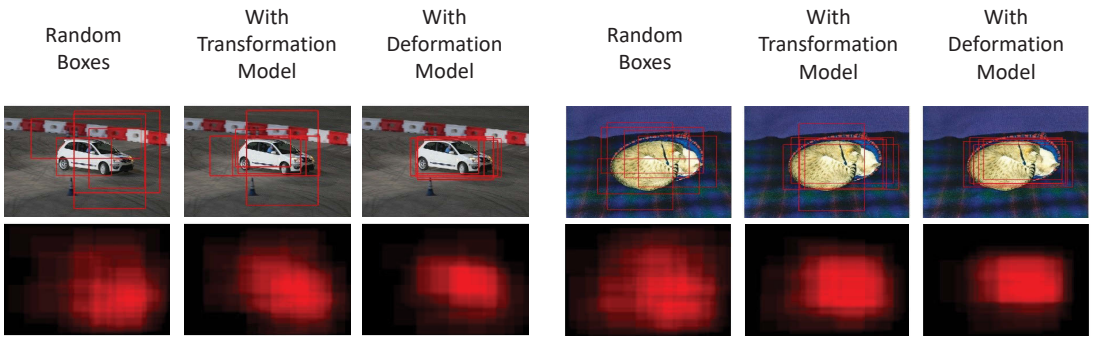


Figure 5.8: Effect of TRM on object coverage by object proposals with failure cases. The man driving the car is missed because his size is too small compared to the car; two cats are identified as one object because one of them is severely occluded by the other.

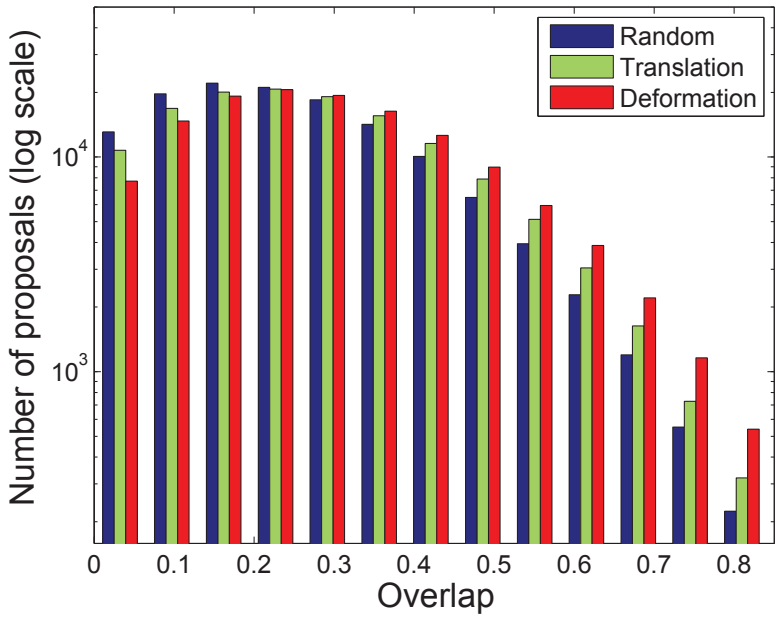


Figure 5.9: Statistical analysis of object coverage by proposals with TRM. The figure shows the statistical analysis of overlap variations of random proposals (blue), translated proposals (green) and deformed proposals (red). By comparing the hists, it is obvious that TRM increase the number of proposals with high overlaps and reduce the number of proposals with low overlaps.

generators including Rahtu [132], Edge Boxes [200], Selective Search [157] and Objectness [2]. The codes of Rahtu and Edge Boxes are set with default parameters, and a few thousands of proposals can be obtained with these methods. For Selective Search, we choose one set of parameters to control the number of proposals at a similar level with Rahtu and Edge Boxes. In addition, for Objectness which generates $\sim 10^5$ ranked proposals per image, we only select 1000 proposals with highest scores to achieve higher efficiency and also maintain a similar number of proposals used for evaluation. In the following parts, these proposals are regarded as original proposals and we perform the TRM on them.

In the second experiment, we compare the mean overlaps between original proposals and transformed ones. The overlap scores between original proposals and ground truth annotations are ranked in descending order and saved in set $\mathcal{O}$. Similarly, the overlap scores of transformed proposals are saved in set $\mathcal{T}$. Afterwards, the mean value of K highest overlap scores in set $\mathcal{O}$ and set $\mathcal{T}$ are calculated. In this sector, K is increased from 10 to 1000 with a step of 5 to illustrate how our model improves the performance of original proposal generators under different proposal quantity requirements. As Fig. 5.10 shows, when compared to above four proposal generators, the mean overlap scores of transformed proposals (dashed lines) are consistently higher than the original ones (solid lines). Consequently, the introduction of the transformation model has been demonstrated to be beneficial for improving the mean overlap scores of the original proposals, which also means that the transformed proposals are more likely to cover objects in diversified scenes in different images.

The third trial compares the mean overlaps and the required number of proposals, when these proposals are produced by original methods and methods boosted by TRM separately, to evaluate the efficiency of TRM. Commonly, the performance of a proposal-generating scheme can be better if more candidate proposals are generated. Using our framework, the number of proposals to achieve roughly the same level of performance should be less than that of original proposal-generating systems. In this trial, we record and plot the mean overlap of the best 100 proposals in Fig. 5.11 as the number of proposals generated by original and transformed schemes grow. Based on the results, it can be found that much fewer proposals are required to achieve similar mean overlap of best

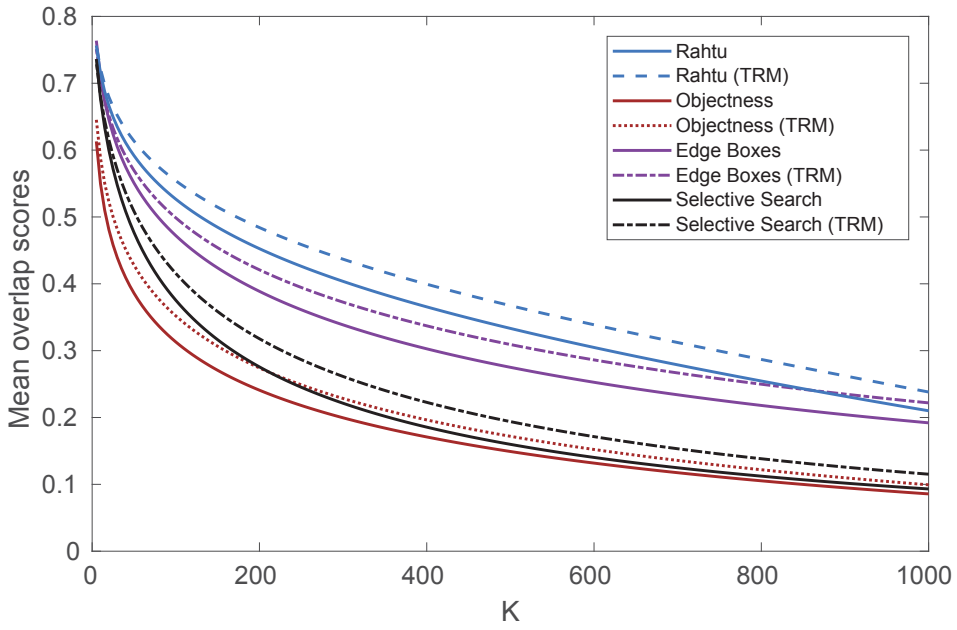


Figure 5.10: Improvements on mean overlap scores (w.r.t. ground truth) when applying TRM on popular object proposal algorithms. The means of K highest overlap scores achieved by original proposals and corresponding transformed proposals are respectively plotted by solid and dashed lines when K varies from 10 to 1000. For each tested proposal generator, it has been shown that the dashed line is always above the solid line, which proves that better coverages on objects can be obtained if proposals are refined by TRM.

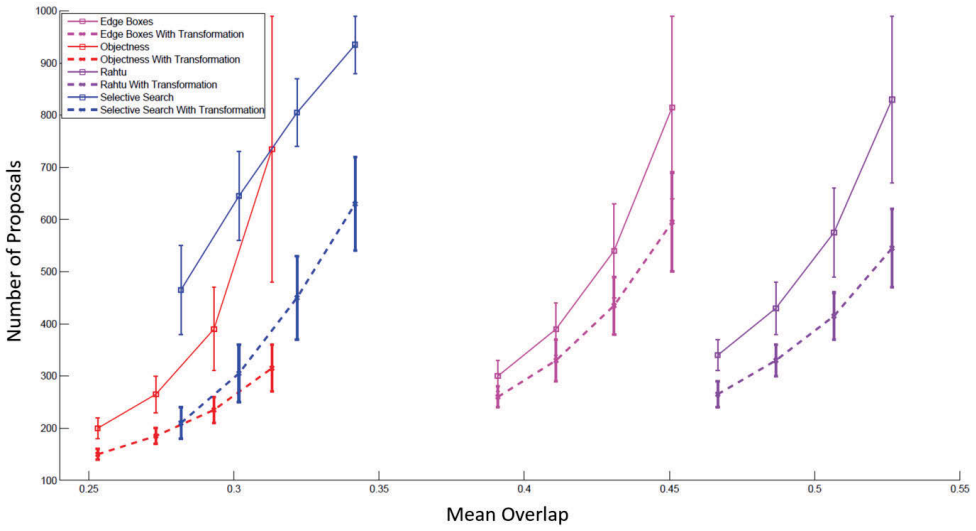


Figure 5.11: Comparison between mean overlaps and a required number of proposals. The transformed proposals (in dashed lines) could achieve the same overlap scores with less number of proposals.

100 proposals by using our transformation model. As an example, the original Edge Box method needs approximately $600 \sim 1000$ proposals to make the mean overlap of its best 100 proposals achieve the level of 0.45, while the transformed Edge Box method only needs $480 \sim 650$ proposal to obtain a similar level of overlap.

In another point of view, the fourth trial compares the numbers of required proposals to achieve similar performance for each evaluated method. In general, the performance of top K windows would be better if more candidate proposals are generated. Under this situation, by introducing transformation models, the number of proposals to achieve roughly the same level of performance is supposed to be much smaller than original schemes. Fig. 5.12 shows the comparisons in the numbers of proposals to achieve similar performance for using original proposals and transformed ones. The boxes in the same column share a similar range of mean overlap of top 100 proposals. Refined proposals sourced from different proposal generators are coloured by different colours. Based on the illustrated results, it has shown that our method requires much fewer proposals to achieve a similar mean overlap score for the top 100 proposals. In particular, in the third

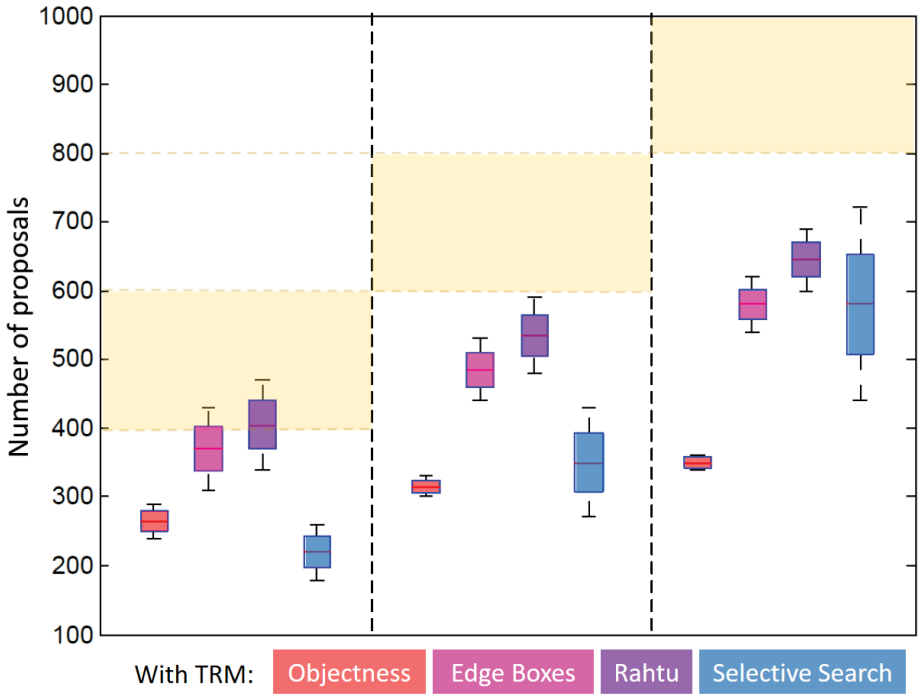


Figure 5.12: This box plot shows the comparisons in the numbers of proposals to achieve similar performance for using original proposals and transformed ones. The boxes in the same column share a similar range of mean overlap of top 100 proposals. Refined proposals sourced from different proposal generators are coloured differently. For Selective Search, to achieve the similar mean overlap score of top 100 windows when $400 \sim 600$ original proposals are generated in total (as the yellow area marks), only $180 \sim 250$ transformed proposals (marked by the blur bar) are required for using TRM to improve the proposals. Thus a fewer number of transformed proposals is required to achieve the similar performance.

column, to achieve the similar mean overlap score of top 100 proposals when $800 \sim 1000$ original proposals are generated in total by Edge Boxes (as the yellow area marks), only $550 \sim 600$ transformed proposals (marked by the magenta bar) are required for using TRM to improve the proposals. As a consequence of the above two experiments, our framework is able to provide high-quality proposals based on fewer total generated proposals and thus can boost the efficiency of original proposal generators.

5.3.2 Compared With Regional Proposal Network

TRM with CNN feature is also trained and test on the VOC 2007 dataset. The generation of training samples is as follows: a set of pre-defined boxes are first generated on the whole image. Then the pre-defined box which has a high IoU value with the ground truth object is regarded as a positive sample. Only positive samples are used for training the TRM, and the regression labels are obtained based on the offsets in [x, y, w, h] regarding the most related ground truth box. Concerning the running time, our approach is end to end and can produce refined proposals simultaneously for an image. If 100 proposals are refined on the image, the extra time for processing this image is $1ms$ and the extra space for parameters is 3MB.

The performance of the TRM based on CNN features is compared with the regional proposal network (RPN) which is widely used for generating proposals. Using a set of pre-defined boxes (known as anchors), RPN predicts which box at each image location may contain an object and then decides how to adjust the predicted box to better cover the object. However, RPN is only implemented by a single convolution operation using the 3x3 kernel. We argue that this is not appropriate to handle anchors with different scales and aspect ratios because the receptive field of the network is too limited. In this section, we can prove the effectiveness of the proposed transformation module to tackle the mentioned issue of RPN based on improvements over proposal recall rates and final detection performance. All the following experiments are conducted based on the VGG network.

5.3.2.1 Improvement On Proposals

For a qualitative comparison, results are shown in Fig. 5.13. The top 3 proposals from TRM are shown with red boxes while the top 3 of RPN are illustrated with dashed blue boxes (Boxes are judged and ranked as in Fast RCNN [54]). Generally, the results given by TRM is more precise and compact. TRM works well with a cluttered background as the 4th row shows or with small objects as listed in the 5th row. In addition, our model performs well on a variety of object categories, including different animals, transportations, and indoor objects, which

have irregular contour, along with huge shape and appearance variations. Besides, the proposed method can handle objects with diverse pose towards the camera. For example, no matter the dog is front to or side to the camera, it can be precisely covered by the top proposals. Failure cases include the cat which is heavily occluded by a desk as the bottom-left image shows, or meaningless area found by the small box in the bottom-right image.

For quantitative comparisons, we evaluate the recall rates of the proposals generated by RPN and the proposed method. Fig. 5.14 shows the experimental results on the VOC 2007 test set. From the figure, we can find that the proposed transformation module effectively increases the recall rates using the same number of generated proposals. Comparing among the sub-figures, it is obvious that our method keeps advantages in the recall rate regardless of the number of proposals. Such advantage is distinct when the number of proposals is small. For example, in Fig. 5.14(a), proposals obtained by the proposed method is able to provide around 20% improvement over the recall rate of RPN. Since higher recall rates mean that more proposals can cover the ground-truth data, our method is demonstrated to have the ability to produce higher quality proposals.

5.3.2.2 Improvement On Detection

In addition to the recall rates of the proposals, we also evaluate detection performance using the proposals generated by RPN and the proposed method. Table 5.1 shows the statistics about the detection score on VOC 2007 test set. Given different numbers of proposals, the evaluated performance illustrates that our method can help the detector achieve better detection score. In specific, our model provides significant improvement for detecting difficult objects such as "boat" and "bottle" that may be extremely slim. Moreover, the overall detection score can also be improved by our method, indicating that the proposal generated by the proposed method is more advantageous for the object detection task. Particularly, when the number of used proposals is extremely small, our method shows great strength in facilitating object detectors to achieve high detection rates. As shown in Table 5.1, if the number of proposals is 50, the mean average precision of our method is 66.2 while the RPN is 63. When the number reduced to 10, our

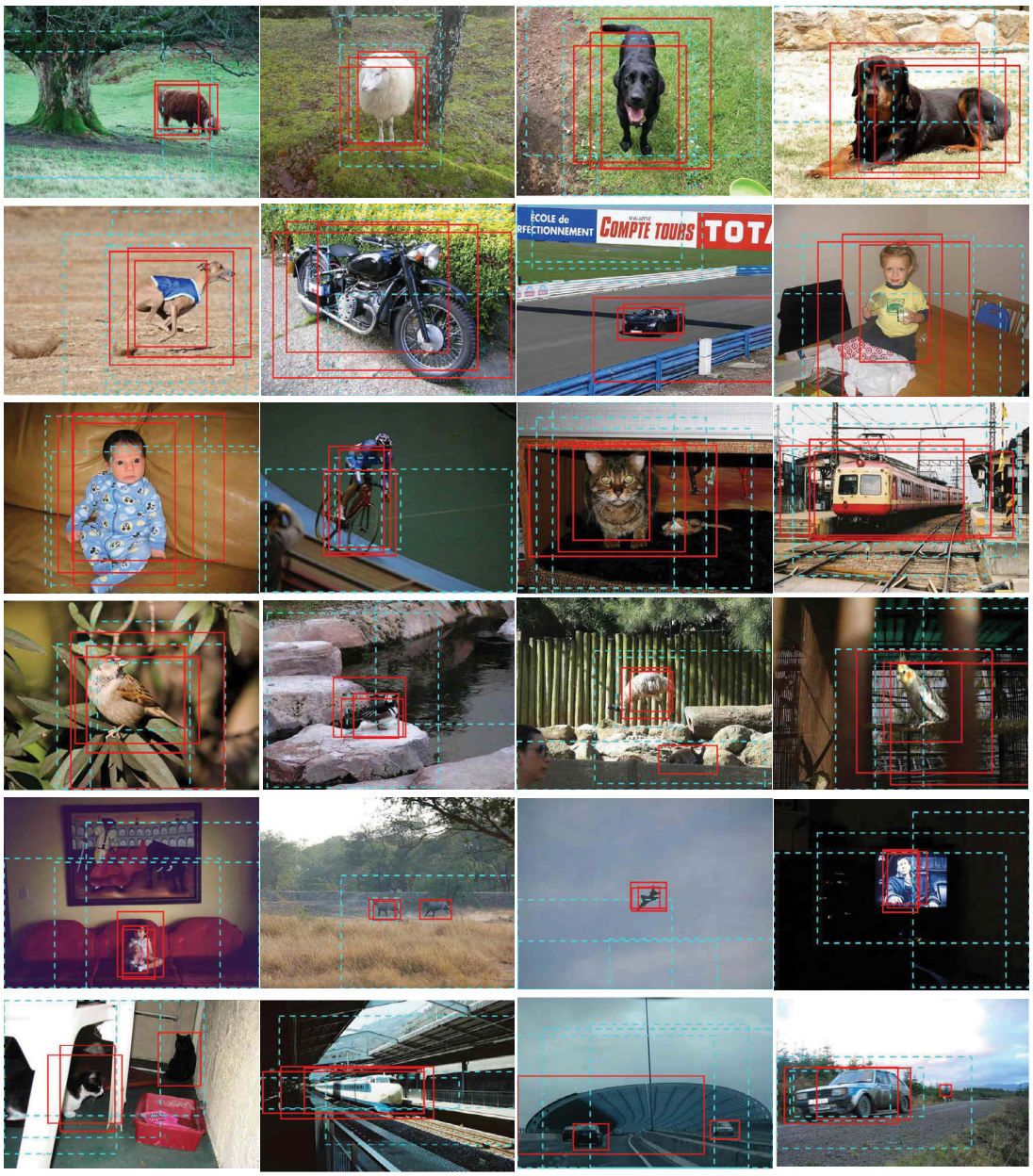


Figure 5.13: Performance comparison between the proposed FoPN and RPN. The red bounding boxes are results of the proposed method while the blue dashed bounding boxes are from RPN.

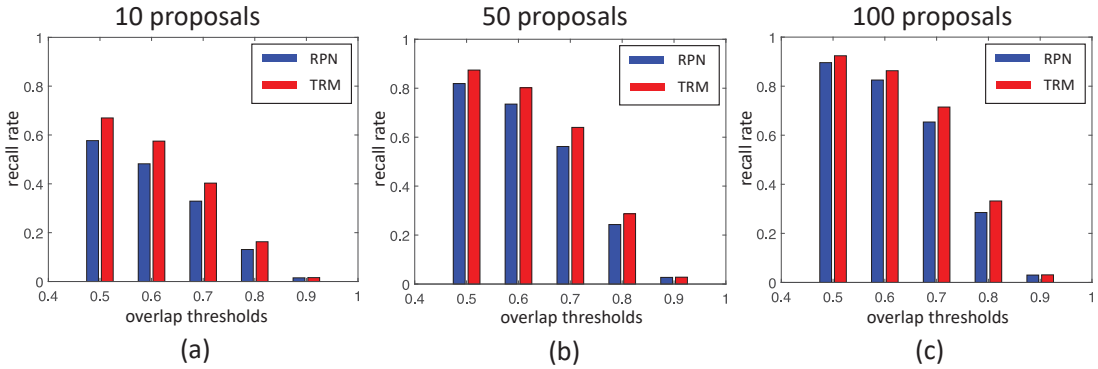


Figure 5.14: The recall rates of the proposed proposal generation method (TRM) compared to the Region Proposal Network (RPN) [133]. The recall rates of the compared methods are evaluated with top 10, 50 and 100 proposals.

Method	Num Prop	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mAP
RPN	10	53.6	54.0	43.9	39.3	24.8	60.8	54.4	62.9	24.0	61.3	51.1	62.2	72.1	51.7	53.0	22.8	52.0	47.3	62.9	35.0	49.5
TRM	10	61.4	54.1	59.6	44.4	33.7	61.6	63.1	79.0	32.3	68.4	40.4	70.4	72.1	60.0	61.6	29.9	60.0	53.8	69.7	52.5	**56.4**
RPN	50	62.3	71.1	61.3	51.0	44.9	69.5	71.9	79.5	38.8	69.2	59.0	77.9	79.9	69.7	69.4	32.8	60.5	55.8	77.6	57.9	63.0
TRM	50	68.7	70.8	67.4	51.8	47.2	76.7	79.3	79.5	44.3	75.1	61.4	77.2	80.1	69.4	70.0	32.7	67.6	61.3	76.7	66.4	**66.2**
RPN	90	**69.4**	71.3	**69.1**	51.1	45.5	70.2	79.5	80.2	44.8	**75.6**	**64.5**	**78.2**	**80.4**	**70.0**	69.5	36.5	67.4	**61.6**	**77.8**	62.6	66.3
TRM	90	67.7	**76.5**	67.0	**55.4**	**50.3**	**77.1**	**79.9**	**85.2**	**48.9**	75.1	63.0	77.2	**80.4**	69.1	**76.1**	**36.7**	**67.6**	60.9	77.2	**70.4**	**68.1**

Table 5.1: Detection score for using different numbers of proposals generated by RPN and the proposed method. The detection is performed using the Fast RCNN method. Best scores for each category and final performance are illustrated in bold.

method has higher precision than RPN on almost all the sub-sequences.

5.3.3 Improvement On Tracking

Apart from improvements on detection, we evaluate the tracking-by-detection performance using the proposals generated by RPN and the proposed method. The test sets come from the Visual Tracker Benchmark[1], which include almost all the challenges and corruptions for the trackers in the benchmark. In specific, these challenges are marked by the benchmark as follows: The *Jogging* sequence contains multiple targets, where a tracker can be attracted by other targets and

[1] `http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html`

cause drifting. Other challenging aspects in this sequence include occlusion, non-rigid object deformation and out-of-plane rotation that the target rotates out of the image plane. The challenging aspects in the *CarScale* sequence include scale variation, occlusion, fast motion, in-plane rotation that the target rotates in the image plane and out-of-plane rotation. The *Couple* sequence contains scale variation, non-rigid object deformation, fast motion, out-of-plane rotation and background clutters. The *Human2* sequence is suffered by illumination variation, scale variation, motion blur and out-of-plane rotation.

A powerful tracker called MDNet, which is the champion of Visual Object Tracking Challenge, is chosen as the baseline tracker in the experiment. In this algorithm, the online tracking is performed by evaluating the candidate windows *randomly sampled* around the previous target state. To evaluate the performance of our model, we replace the randomly generated windows with proposals output by the proposed TRM model to construct an enhanced tracker called TRM Tracker (TRMT). To further outstand the contribution of TRM, we also compare the tracking performance with RPN tracker, where the MDNet is refined by the Regional Proposal Network (RPN). RPN is a CNN based method that predicts which box at each image location may contain an object and then decides how to adjust the predicted box to better cover the object. The number of proposals of each test method is equal, which is about tens of windows. Both qualitative comparisons and quantitatively comparisons are presented in this part.

Firstly, the qualitative comparison, i.e., the visualized results on each test sequence, are shown by selected frames in each test sequences. For each sub-figure, the frame number is denoted at the top-left corner, and the results from a mix of methods are marked by bounding boxes in various colours. The tracking results of the baseline are marked in black. The result of RPN tracker is marked in blue and the result of our method is marked by red. Fig. 5.15 illustrates the result on the *CarScale* sequence. The first row shows that the result of the baseline and RPN is not accurate, especially in size and dimension. When occlusion appears, as frame 161 in the second-row shows, the bounding box of our method tightly covers the target while other results are not accurate. When in-plane rotation happens, as the third-row shows, the appearance of the target greatly changed. But our tracker can still track the car accurately. Fig. 5.16 shows a couple walks

across the street. In this test set, the camera moves when recording the video. In addition, the background contains different kinds of objects which may distract the tracker. As shown in the frame 73 and 79 listed in the second row, although the RPN tracker is drifted to the car besides the couple, our tracker keeps on the target and our tracking results are more stable than others. In Fig. 5.17, trackers can be distracted by a similar object, the woman in white shorts, especially when occlusion appears in frame 82. In addition, the baseline tracker loses the target in frame 305, but our tracker performs robustly in this sequence. The *Human2* sequence is challenging since the target suffers severe illumination change and present huge posture change, like squat down and jump. Fig. 5.18 shows that our method is robust against occlusion, as shown in frame 248, 348, and 640. It is also robust against appearance change. No matter the human is in front of, side of, or back to the camera, our tracker catches him.

Figure 5.15: The tracking result on *CarScale*. The results of the baseline method are marked in black. The results of RPN are in blue and ours TRMT are in red (best viewed in colour).

Secondly, the statistical comparisons among the baseline, RPN, and the proposed method are provided. The overlap score on test sequences are shown by Figure 5.19, 5.20, 5.21 and 5.22. In these figures, the results of baseline, RPN, and ours are marked in black, blue and red correspondingly. There are six plots in each figure. Three of them are in light colour. These light plots present the overlap score between outputs and the ground truth of each frame. The smoothed overlap score is shown by the plots in a darker colour. In the *CarScale* sequence, the baseline failed (the overlap rate is smaller than 0.5) after frame 100, but RPN and ours successfully track the target from the beginning to the end. The overlap

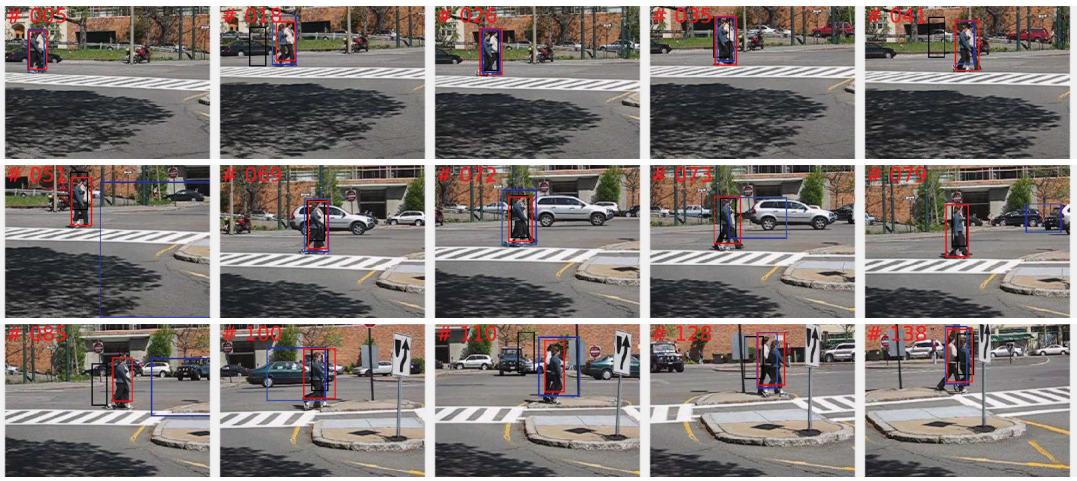


Figure 5.16: The tracking result on *Couple*. The results of the baseline method are marked in black. The results of RPN are in blue and ours TRMT are in red (best viewed in colour).

Sequence	CarScale	Jogging	Couple	Human2	Average
Baseline	0.4015	0.2553	0.4345	0.3028	0.3485
RPN	0.6291	0.3590	0.5151	0.5585	0.5154
Ours (TRMT)	**0.7849**	**0.6343**	**0.6582**	**0.5981**	**0.6689**

Table 5.2: Mean overlap rate. Best scores are illustrated in bold.

rate of our method is higher than RPN. In the *Couple* sequence, the performance of baseline tracker and RPN tracker is similar while ours are better than them. In the *Human2* sequence, the overlap rate various during time with the appearance suffers a variety of challenges. On average, our method outperforms RPN, and is much better than the baseline. In addition, on the *Jogging* set, our performance is still better than the others.

Thirdly, the mean overlap rate on the whole sequence is shown by the table 5.2. The proposed method has the highest mean overlap rate on all the test sequences, which proves that our tracker performs better than the others.

Finally, the speed of these methods is discussed in table 5.3. It shows the number of frames processed by each method in one second. Compared with the baseline tracker, although our tracker introduced additional processing scheme

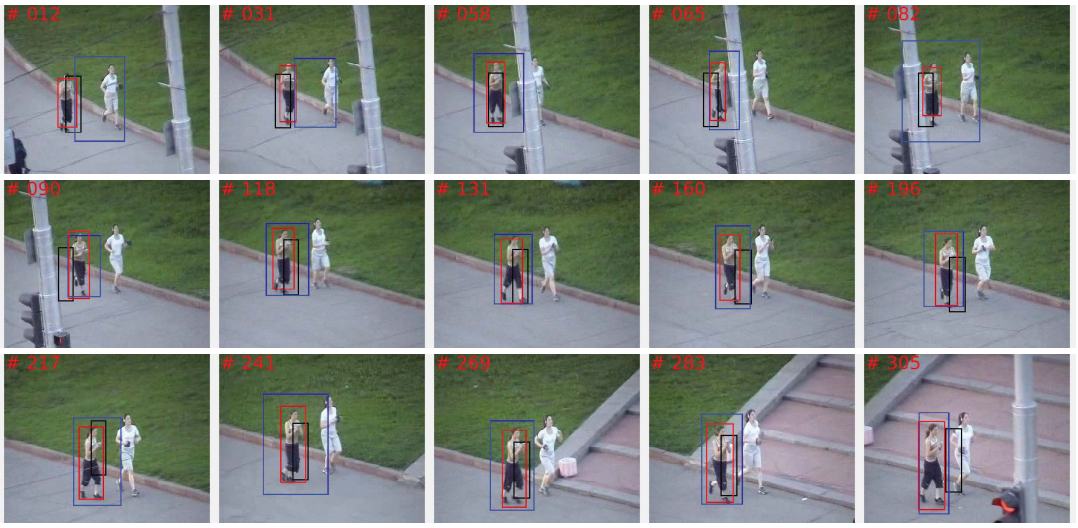


Figure 5.17: The tracking result on *Jogging*. The results of the baseline method are marked in black. The results of RPN are in blue and ours TRMT are in red (best viewed in colour).

Sequence	CarScale	Jogging	Couple	Human2	Average
Baseline	1.3415	6.2583	5.0268	1.3415	3.4920
RPN	2.3544	3.6229	5.4716	2.3544	3.4508
Ours (TRMT)	2.1833	4.1047	5.3632	2.1833	3.4586

Table 5.3: Mean frame rate per second.

on random windows to improve the quality of candidate windows, the running speed is at a similar level with the baseline tracker.

5.4 Conclusion

In this thesis, we present the TRM model which improves the quality of object candidate windows for visual trackers using the tracking-by-detection strategy. By analysing the context, the TRM adjusts the location and shape of the proposal to gradually focus the proposal on the target area. The model is orthogonal to existing proposal-generating schemes and can be applied to proposals generated with hand-crafted features or CNN features. The adjustments are efficient

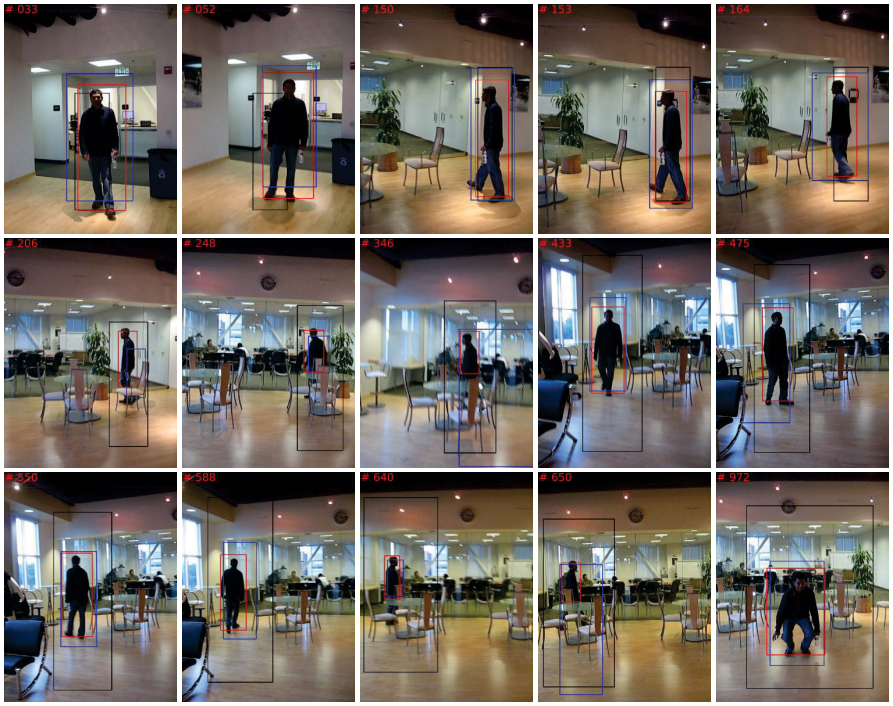


Figure 5.18: The tracking result on *Human2*. The results of the baseline method are marked in black. The results of RPN are in blue and ours TRMT are in red (best viewed in colour).

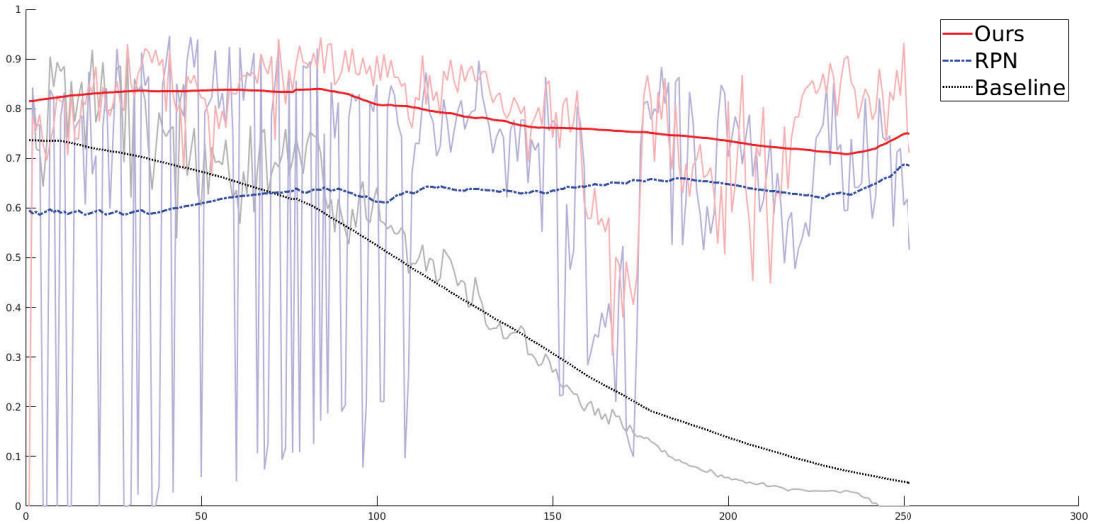


Figure 5.19: The overlap rate on *CarScale*.

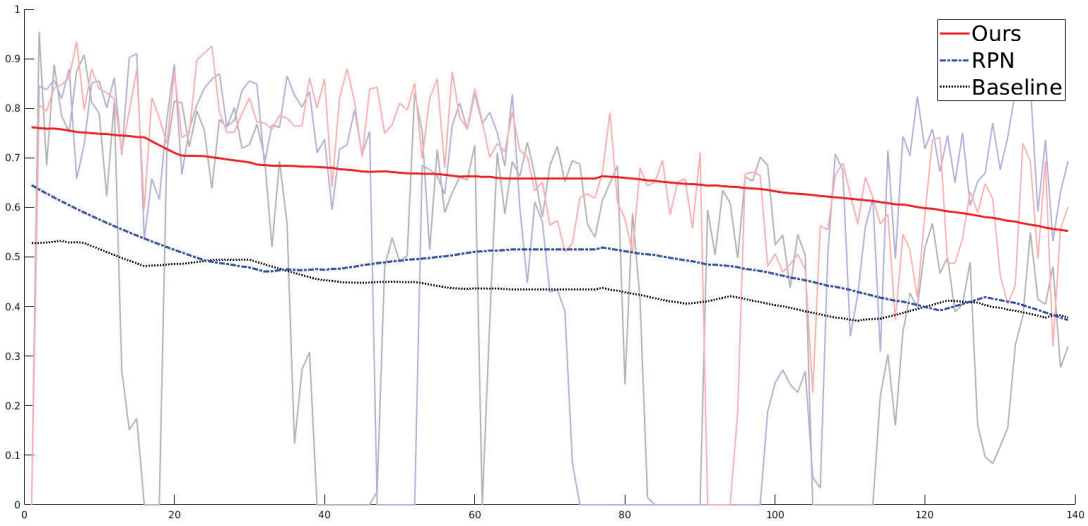


Figure 5.20: The overlap rate on *Couple*.

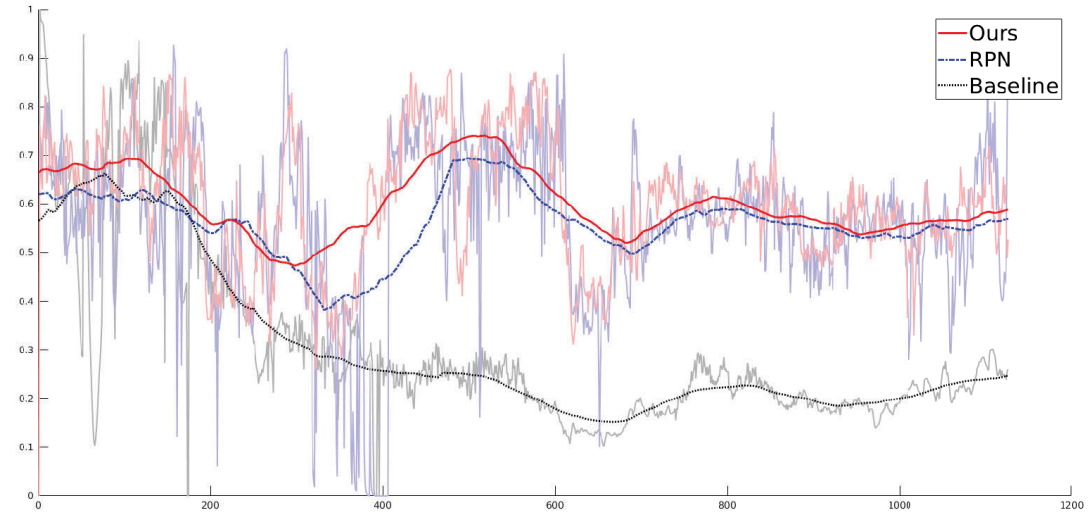


Figure 5.21: The overlap rate on *Human2*.

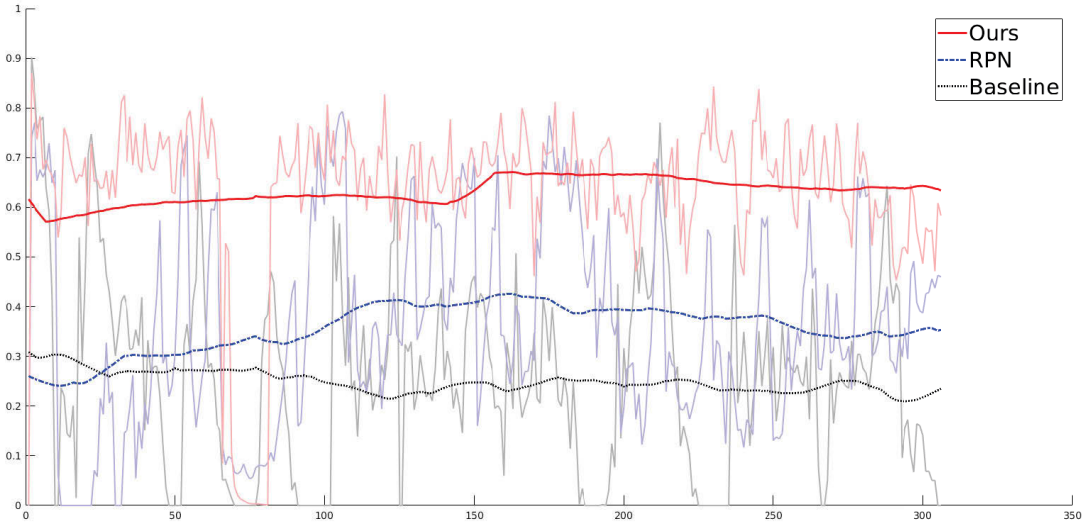


Figure 5.22: The overlap rate on *Jogging*.

to compute and effective. Our experiments show that, compared with the original proposals generated by existing proposal generators, the transformed proposals can reach improved coverage with objects or maintain the performance of the original proposals with fewer numbers of them. In addition, the model can be easily embedded into an existing object detection or tracking framework to boost the performance of the detector or tracker. In the future, our method can be used to improve the performance of other visual applications, such as image segmentation, multi-object tracking [66], as well as other image processing techniques.

Algorithm 2 Deformation Algorithm

Input: Proposal window $\mathbf{b}$, MaxRound, MaxStep
Output: New proposal window $\mathbf{b}'$

for k in $\{1, \cdots,$ MaxRound$\}$ **do**
  for direction in {horizontal, vertical} **do**
    $\mathbf{b_{sh}} \leftarrow$ shrink $\mathbf{b_{k-1}}$ in direction by Δ
    $\mathbf{b_{st}} \leftarrow$ stretch $\mathbf{b_{k-1}}$ in direction by Δ
    if $C(\mathbf{b_{sh}}) > C(\mathbf{b_{k-1}}) + \lambda$ **then**
      $operation \leftarrow$ shrink
      $s_1 := C(\mathbf{b_{sh}})$
      $\mathbf{b_{new}} := \mathbf{b_{sh}}$
    else if $C(\mathbf{b_{st}}) > C(\mathbf{b_{k-1}}) + \lambda$ **then**
      $operation \leftarrow$ stretch
      $s_1 := C(\mathbf{b_{st}})$
      $\mathbf{b_{new}} := \mathbf{b_{st}}$
    else
      $\mathbf{b_{new}} := \mathbf{b_{k-1}}$
    end if
    $step := 0$
    repeat
      $step := step + 1$
      $s_0 := s_1$
      $\mathbf{b_{new}} \leftarrow operation(\mathbf{b_{new}})$
      $s_1 := C(\mathbf{b_{new}})$
    until $s_1 < s_0 + \lambda$ **or** $step \geq$ MaxStep
    $\mathbf{b_k} := \mathbf{b_{new}}$
  end for
end for
$\mathbf{b}' := \mathbf{b_{MaxRound}}$

Chapter 6

Conclusions

Over the past decades, visual object tracking has long been an important computer vision task for researchers. Although various methods have been proposed and promising progress has been made, it is still challenging for researchers to build a robust and efficient tracking model due to various challenging issues, including occlusion, deformation, shape variation, scale change, background clutter, and so on. By addressing these issues, this thesis aims to incorporate visual contexts to achieve robust visual tracking. Considering contexts in a visual processing system is natural and reasonable. Evidence from both the neurophysiological and statistical properties of perceiving typical natural scenes in the human brain supports the view that humans utilize the spatial and temporal context when recognizing objects. Accordingly, it could be advantageous to utilise visual contexts in the visual tracking process. On the one hand, spatial context can provide complementary visual cues when tracking an object. For example, if the target is occluded by other objects, spatial context can enable a tracker to infer about the actual state of this target without accessing its complete appearance. This can further reduce the risk of losing the target in the following frames. On the other hand, temporal context can provide strong clues about where an object should be in each new frame. Different from other computer vision tasks such as generic object detection and semantic segmentation that make predictions on still images, objects in a video present slight appearance changes in consecutive frames, thus the information from previous frames could be helpful to identify the state of the target in the current frame. In particular, by incorporating the spatial and

temporal context, this thesis designs three robust visual tracking algorithms to tackle the tracking problem. The proposed tracking algorithms include a tracker called MMST that is robust against occlusion, a Bi-channel FCN tracker specialized at tackling shape variation, and a tracker providing candidate windows that better cover the target.

In **MMST** (Chapter 3), a robust and efficient visual object tracking method has been developed based on an improved subspace learning-based appearance model. In this appearance model, we introduce the novel mask templates that contain the temporal context, greatly reducing the complexity of the system. For the proposed model, we provide a theoretical guarantee of the efficiency of the solution. Our method is also characterized by its exploitation of the temporal context, called 'dynamic information' of the tracking target in the study, in nearby frames. The temporal context can significantly improve the tracking accuracy and coverage of the target. Extensive experiments validated the efficiency and robustness of our method, especially in situations with frequent and obvious large-scale corruptions, such as occlusions and illumination variations. Our MMST model could also be extended to multi-object tracking tasks to enhance the association approaches when estimating the states of a mix of objects.

Based on Bi-channel FCN, we present **FCN^2** (Chapter 4) for visual object tracking that 1) produces finer tracking results at the pixel level, and 2) works for the generic object without fitting the network to the appearance of any specific object class which needs a large scale of training data. The proposed bi-channel FCN progressively incorporates spatial context information of a higher level into features at a lower level. Furthermore, the proposed model can also extract the temporal context from the previous frame. In general, the introduced contexts work together with optical flow information to produce a robust tracking result. The experiment results demonstrate that the proposed algorithm achieves superior tracking performance, especially in tackling challenges with in-plane rotation and deformation.

Using the tracking-by-detection strategy, we also proposed a tracker called **TRMT** (Chapter 5). In the proposed tracker, a novel transformation model that can help the target-specific online detector better focus on the target is designed to help the candidate windows cover the target more precisely. By analysing the

spatial context around candidate windows, both the location and dimension of each window are refined in an iterative manner. In particular, the proposed TRM model is flexible. It can be combined with methods using hand-crafted features and methods using CNN features. Empirical results prove that the time cost of the proposed TRM model is negligible, while the performance of the enhanced tracker can be boosted. TRMT is robust against object appearance variation and can avoid drifting due to the presence of similar objects.

In the future, the spatial context will be investigated in a long-short range manner to build a robust tracker that can provide pixel-level tracking results. The spatial contexts in both nearby small areas and distanced image patches are stored in a unified form. In this way, a tracker can better separate the target from its surroundings. In addition, extensions of the deep neural network-based tracking algorithms that can encode the multi-view feature to facilitate tracking will be explored.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2006, pp. 798–805. 10, 40

[2] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2189–2202, 2012. 25, 68, 71, 75, 77, 82, 86

[3] R. C. Atkinson and R. M. Shiffrin, "Human memory: A proposed system and its control processes," *Psychology of learning and motivation*, vol. 2, pp. 89–195, 1968. 19

[4] S. Avidan, "Support vector tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004. 16

[5] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 983–990. 10, 39

[6] ——, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, 2011. 10

[7] Y. Bai and M. Tang, "Robust tracking via weakly supervised ranking svm," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1854–1861. 16

[8] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1830–1837. 13, 15, 35, 38, 39, 61

[9] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: Complementary learners for real-time tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1401–1409. 19

[10] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision Workshops*, 2016, pp. 850–865. 3, 23

[11] A. Bibi and B. Ghanem, "Multi-template scale-adaptive kernelized correlation filters," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 50–57. 19

[12] M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in *Proceedings of the European Conference on Computer Vision*. Springer, 2008, pp. 2–15. 25

[13] V. N. Boddeti and B. V. Kumar, "A framework for binding and retrieving class-specific information to and from image patterns using correlation filters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 9, pp. 2064–2077, 2013. 16

[14] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 2544–2550. 18

[15] D. S. Bolme, Y. M. Lui, B. A. Draper, and J. R. Beveridge, "Simple real-time human detection using a single correlation filter," in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*. IEEE, 2009, pp. 1–8. 16

[16] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017. 27, 28, 55

[17] D. Casasent and R. Patnaik, "Analysis of kernel distortion-invariant filters," in *Intelligent Robots and Computer Vision XXV: Algorithms, Techniques, and Active Vision*, vol. 6764. International Society for Optics and Photonics, 2007, p. 67640Y. 11

[18] L. Cehovin, M. Kristan, and A. Leonardis, "An adaptive coupled-layer visual model for robust visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 1363–1370. 11

[19] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." *IEEE transactions on pattern analysis and machine intelligence*, 2017. 78

[20] Z. Chen and Z. Chen, "Rbnet: A deep neural network for unified road and road boundary detection," in *Proceedings of the International Conference on Neural Information Processing*. Springer, 2017, pp. 677–687. 20

[21] Z. Chen, Z. Hong, and D. Tao, "An experimental survey on correlation filter-based tracking," *arXiv preprint arXiv:1509.05520*, 2015. 10, 16

[22] Z. Chen, J. Li, Z. Chen, and X. You, "Generic pixel level object tracker using bi-channel fully convolutional network," in *Proceedings of the International Conference on Neural Information Processing*. Springer, 2017, pp. 666–676. 27

[23] Z. Chen, X. You, B. Zhong, J. Li, and D. Tao, "Dynamically modulated mask sparse tracking," *Cybernetics, IEEE Transactions on*, vol. 47, no. 11, pp. 3706–3718, 2017. 22

[24] Z. Chen, X. You, and J. Li, "Learning to focus for object proposals," in *Proceedings of the IEEE International Conference on Security, Pattern Analysis, and Cybernetics*. IEEE, 2017, pp. 439–444. 24

[25] T.-J. Chin and D. Suter, "Incremental kernel principal component analysis," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1662–1674, 2007. 12

[26] J. Choi, H. J. Chang, S. Yun, T. Fischer, Y. Demiris, J. Y. Choi *et al.*, "Attentional correlation filter network for adaptive visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2017. 19

[27] J. Choi, H. J. Chang, T. Fischer, S. Yun, K. Lee, J. Jeong, Y. Demiris, and J. Y. Choi, "Context-aware deep feature compression for high-speed visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 479–488. 21

[28] J. Choi, H. Jin Chang, J. Jeong, Y. Demiris, and J. Young Choi, "Visual tracking using attention-modulated disintegration and integration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4321–4330. 19

[29] R. T. Collins, "Mean-shift blob tracking through scale space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2003, pp. II–234. 39

[30] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 564–577, 2003. 39

[31] Z. Cui, S. Xiao, J. Feng, and S. Yan, "Recurrently target-attending tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1449–1458. 22

[32] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3150–3158. 55

[33] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 886–893. 25

[34] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 21–26. 21

[35] M. Danelljan, G. Hager, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014. 18, 61

[36] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1561–1575, 2017. 18

[37] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 58–66. 20

[38] ——, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4310–4318. 18

[39] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 472–488. 21

[40] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2014, pp. 1090–1097. 19

[41] E. R. Davies, *Computer and machintare vision: theory, algorithms, practicalities*. Academic Press, 2012. 1

[42] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 1177–1184. 5, 10, 24, 39

[43] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2013, pp. 1841–1848. 25

[44] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766. 58

[45] S. Duffner and C. Garcia, "Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2480–2487. 55

[46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 81

[47] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010. 82

[48] G. Felsen, J. Touryan, and Y. Dan, "Contextual modulation of orientation tuning contributes to efficient processing of natural stimuli," *Network: Computation in Neural Systems*, vol. 16, no. 2-3, pp. 139–149, 2005. 5

[49] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 2241–2248. 25

[50] D. J. Field, A. Hayes, and R. F. Hess, "Contour integration by the human visual system: evidence for a local association field," *Vision research*, vol. 33, no. 2, pp. 173–193, 1993. 5

[51] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 21–26. 18

[52] H. K. Galoogahi, T. Sim, and S. Lucey, "Multi-channel correlation filters," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2013, pp. 3072–3079. 15

[53] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 188–203. 16

[54] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448. 26, 67, 90

[55] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 580–587. 25

[56] S. Gladh, M. Danelljan, F. S. Khan, and M. Felsberg, "Deep motion features for visual tracking," in *IEEE International Conference on Pattern Recognition*. IEEE, 2016, pp. 1243–1248. 21

[57] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org. 69

[58] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proceedings of the British Machine Vision Conference*, vol. 1, no. 5, 2006, p. 6. 15, 39

[59] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proceedings of the European Conference on Computer Vision*. Springer, 2008, pp. 234–247. 39

[60] H. Grabner, J. Matas, L. Van Gool, and P. Cattin, "Tracking the invisible: Learning where the object might be," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1285–1292. 24

[61] R. M. Gray, "Toeplitz and circulant matrices: A review," *Communications and Information Theory*, vol. 2, no. 3, pp. 155–239, 2005. 18

[62] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016. 26

[63] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr, "Struck: Structured output tracking with kernels," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2096–2109, 2016. 17, 61

[64] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 263–270. 10, 15, 39

[65] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold siamese network for real-time object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4834–4843. 24

[66] Z. He, X. Li, X. You, D. Tao, and Y. Y. Tang, "Connected component model for multi-object tracking," *IEEE transactions on image processing*, vol. 25, no. 8, pp. 3698–3711, 2016. 100

[67] Z. He, S. Yi, Y.-M. Cheung, X. You, and Y. Y. Tang, "Robust object tracking via key patch sparse representation," *Cybernetics, IEEE Transactions on*, vol. 47, no. 2, pp. 354–364, 2017. 67

[68] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 749–765. 23

[69] J. F. Henriques, J. Carreira, R. Caseiro, and J. Batista, "Beyond hard negative mining: Efficient detector learning via block-circulant decomposition," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2013, pp. 2760–2767. 18

[70] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of the European Conference on Computer Vision*. Springer, 2012, pp. 702–715. 3, 5, 17, 18, 39, 61

[71] ——, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015. 5, 18

[72] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 749–758. 5, 19

[73] Z. Hong, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Tracking using multilevel quantizations," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 155–171. 15

[74] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8. 25

[75] Y. Hua, K. Alahari, and C. Schmid, "Occlusion and motion reasoning for long-term tracking," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 172–187. 14

[76] H. Izadinia, I. Saleemi, W. Li, and M. Shah, "2t: Multiple people multiple parts tracker," in *Proceedings of the European Conference on Computer Vision*. Springer, 2012, pp. 100–114. 11

[77] S. D. Jain, B. Xiong, and K. Grauman, "Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 2117–2126. 27

[78] K.-H. Jeong, P. P. Pokharel, J.-W. Xu, S. Han, and J. C. Principe, "Kernel based synthetic discriminant function for object recognition," in *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*, vol. 5. IEEE, 2006, pp. V–V. 18

[79] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1822–1829. 11

[80] Z. Kalal, J. Matas, and K. Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 49–56. 10

[81] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, 2012. 10, 39

[82] F. S. Khan, J. van de Weijer, and M. Vanrell, "Modulating shape features by color attention for object recognition," *International Journal of Computer Vision*, vol. 98, no. 1, pp. 49–64, 2012. 26

[83] H. Kiani Galoogahi, T. Sim, and S. Lucey, "Correlation filters with limited boundaries," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4630–4638. 18

[84] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder, "The visual object tracking vot2015 challenge results," in *IEEE International Conference on Computer Vision Workshops*, 2015, pp. 1–23. 19

[85] M. Kristan, J. Matas, A. Leonardis, T. Vojíř, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 11, pp. 2137–2155, 2016. 21

[86] B. V. Kumar, J. A. Fernandez, A. Rodriguez, and V. N. Boddeti, "Recent advances in correlation filter theory and application," in *SPIE Defense+ Security*. International Society for Optics and Photonics, 2014, pp. 909 404–909 404. 16

[87] S. Kwak, W. Nam, B. Han, and J. H. Han, "Learning occlusion with likelihoods for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 1551–1558. 11

[88] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1269–1276. 3, 40

[89] ——, "Tracking by sampling trackers," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 1195–1202. 40

[90] K. Lebeda, S. Hadfield, J. Matas, and R. Bowden, "Long-term tracking through failure cases," in *IEEE International Conference on Computer Vision Workshops, 2013 IEEE International Conference on*. IEEE, 2013, pp. 153–160. 14

[91] C. Leistner, A. Saffari, P. M. Roth, and H. Bischof, "On robustness of online boosting-a competitive study," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*. IEEE, 2009, pp. 1362–1369. 15

[92] D. Levi, N. Garnett, E. Fetaya, and I. Herzlyia, "Stixelnet: A deep convolutional network for obstacle detection and road segmentation." in *The British Machine Vision Conference*, 2015, pp. 109–1. 55

[93] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8971–8980. 24

[94] F. Li, Y. Yao, P. Li, D. Zhang, W. Zuo, and M.-H. Yang, "Integrating boundary and center correlation filters for visual tracking with aspect ratio variation." 19

[95] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 4, p. 58, 2013. 10, 11

[96] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 254–265. 3, 18

[97] Y. Li, J. Zhu, and S. C. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 353–361. 18

[98] H. Lim, O. I. Camps, M. Sznaier, and V. I. Morariu, "Dynamic appearance modeling for human tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2006, pp. 751–757. 12

[99] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and k-selection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 1313–1320. 39

[100] T. Liu, G. Wang, L. Wang, and K. L. Chan, "Visual tracking via temporally smooth sparse coding," 2012. 11

[101] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4902–4912. 18

[102] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 21–37. 26, 68

[103] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017. 26

[104] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440. 55, 56, 59

[105] S. Lucey, "Enforcing non-positive weights for stable support vector tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8. 16

[106] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3074–3082. 20, 61

[107] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 5388–5396. 5, 19

[108] A. Mahalanobis, B. Vijaya Kumar, S. Song, S. Sims, and J. Epperson, "Unconstrained correlation filters," *Applied Optics*, vol. 33, no. 17, pp. 3751–3759, 1994. 16

[109] T. Malisiewicz, A. Gupta, A. Efros *et al.*, "Ensemble of exemplar-svms for object detection and beyond," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 89–96. 25

[110] K. Matej, J. Matas, A. Leonardis, M. Felsberg, L. Čehovin, G. Fernández, T. Vojíř, G. Häger, G. Nebehay, R. Pflugfelder *et al.*, "The visual object tracking vot2015 challenge results," in *Workshop on the Visual Object Tracking Challenge (VOT, in conjunction with IEEE International Conference on Computer Vision)*. IEEE, 2015. 3

[111] X. Mei and H. Ling, "Robust visual tracking using l1 minimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1436–1443. 34, 35

[112] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient l1 tracker with occlusion detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 1257–1264. 3, 35, 39, 40

[113] X. Mei, H. Ling, Y. Wu, E. P. Blasch, and L. Bai, "Efficient minimum error bounded particle resampling l1 tracker with occlusion detection," *Image Processing, IEEE Transactions on*, vol. 22, no. 7, pp. 2661–2675, 2013. 13, 14, 28

[114] D. Melcher, "Predictive remapping of visual features precedes saccadic eye movements," *Nature neuroscience*, vol. 10, no. 7, pp. 903–907, 2007. 68

[115] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International journal of computer vision*, vol. 60, no. 1, pp. 63–86, 2004. 25

[116] A. Milan, L. Leal-Taixé, K. Schindler, and I. Reid, "Joint tracking and segmentation of multiple targets," in *Proceedings of the IEEE Conference*

on Computer Vision and Pattern Recognition. IEEE, 2015, pp. 5397–5406. 1

[117] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems*, 2014, pp. 2204–2212. 27

[118] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1396–1404. 18

[119] S. A. Mulay, P. Devale, and G. Garje, "Intrusion detection system using support vector machine and decision tree," *International Journal of Computer Applications*, vol. 3, no. 3, pp. 40–43, 2010. 25

[120] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4293–4302. 22

[121] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Advances in Neural Information Processing Systems*. MIT Press, 2002, pp. 841–848. 11

[122] J. Ning, J. Yang, S. Jiang, L. Zhang, and M.-H. Yang, "Object tracking via dual linear structured svm and explicit feature map," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4266–4274. 17

[123] B. A. Olshausen, C. H. Anderson, and D. C. Van Essen, "A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information," *The Journal of Neuroscience*, vol. 13, no. 11, pp. 4700–4719, 1993. 27

[124] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1940–1947. 39

[125] D. Pathak, R. Girshick, P. Dollar, T. Darrell, and B. Hariharan, "Learning features by watching objects move," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2017. 27

[126] R. Patnaik and D. Casasent, "Fast fft-based distortion-invariant kernel filters for general object recognition," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2009, pp. 725 202–725 202. 17

[127] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 20, 27

[128] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 724–732. 3, 60, 61

[129] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proceedings of the European Conference on Computer Vision*. Springer, 2002, pp. 661–675. 10, 39

[130] F. Pernici and A. Del Bimbo, "Object tracking by oversampling local features," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 12, pp. 2538–2551, 2014. 13

[131] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4303–4311. 20

[132] E. Rahtu, J. Kannala, and M. Blaschko, "Learning a category independent object detection cascade," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 1052–1059. 25, 26, 68, 86

[133] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural*

information processing systems, 2015, pp. 91–99. xiv, 1, 24, 26, 55, 68, 80, 93

[134] R. Rifkin, G. Yeo, and T. Poggio, "Regularized least-squares classification," *Nato Science Series Sub Series III Computer and Systems Sciences*, vol. 190, pp. 131–154, 2003. 15

[135] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008. 10, 12, 39

[136] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 2564–2571. 25

[137] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*. IEEE, 2009, pp. 1393–1400. 15

[138] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "Prost: Parallel robust online simple tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 723–730. 10

[139] M. Savvides and B. Kumar, "Efficient design of advanced correlation filters for robust distortion-tolerant face recognition," in *IEEE Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2003, pp. 45–52. 17

[140] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001. 18

[141] O. Schwartz, A. Hsu, and P. Dayan, "Space and time in visual context," *Nature Reviews Neuroscience*, vol. 8, no. 7, p. 522, 2007. 5

[142] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *International Conference on Learning Representations*. CBLS, April 2014. 25

[143] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1910–1917. 39

[144] F. Shahbaz Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez, "Color attributes for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3306–3313. 19

[145] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1815–1821. 11

[146] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014. 3, 10

[147] Y. Song, C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. Lau, and M.-H. Yang, "Vital: Visual tracking via adversarial learning," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 23

[148] C. Sun, H. Lu, and M.-H. Yang, "Learning spatial-aware regressions for visual tracking," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 21

[149] C. Sun, D. Wang, H. Lu, and M.-H. Yang, "Correlation tracking via joint discrimination and reliability learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 489–497. 18

[150] J. Supancic and D. Ramanan, "Self-paced learning for long-term tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2013, pp. 2379–2386. 14

[151] M. Tang and J. Feng, "Multi-kernel correlation filter for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3038–3046. 19

[152] Y. Tang, N. Srivastava, and R. R. Salakhutdinov, "Learning generative models with visual attention," in *Advances in Neural Information Processing Systems*, 2014, pp. 1808–1816. 27

[153] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1420–1429. 3, 23, 61

[154] M. Tian, W. Zhang, and F. Liu, "On-line ensemble svm for robust object tracking," *Proceedings of the Asian Conference on Computer Vision*, pp. 355–364, 2007. 16

[155] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2015, pp. 4489–4497. 1, 55

[156] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," in *SIAM Journal on Optimization*, 2008. 38, 39

[157] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013. 25, 26, 70, 86

[158] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *Image Processing, IEEE Transactions on*, vol. 18, no. 7, pp. 1512–1523, 2009. 19

[159] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2001, pp. I–511. 15

[160] J. Wang, C. Lu, M. Wang, P. Li, S. Yan, and X. Hu, "Robust face recognition via adaptive sparse representation," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2368–2378, 2014. 13

[161] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3119–3127. 22

[162] ——, "Stct: Sequentially training convolutional networks for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1373–1381. 22

[163] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in neural information processing systems*, 2013, pp. 809–817. 22, 23

[164] Q. Wang, J. Gao, and Y. Yuan, "Embedding structured contour and location prior in siamesed fully convolutional networks for road detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 230–241, 2018. 26

[165] Q. Wang, Y. Yuan, and P. Yan, "Visual saliency by selective contrast," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 7, pp. 1150–1155, 2013. 25

[166] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. 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*, 2018, pp. 4854–4863. 24

[167] Q. Wang, F. Chen, W. Xu, and M.-H. Yang, "Object tracking via partial least squares analysis," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4454–4465, 2012. 12

[168] ——, "Object tracking with joint optimization of representation and classification," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 25, no. 4, pp. 638–650, 2015. 12

[169] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *Proceedings of the International Conference on Computer Vision*. IEEE, 2009, pp. 32–39. 25

[170] X. Wang, G. Hua, and T. X. Han, "Discriminative tracking by metric learning," in *Proceedings of the European Conference on Computer Vision*. Springer, 2010, pp. 200–214. 11

[171] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2013, pp. 17–24. 25

[172] L. Wen, Z. Cai, Z. Lei, D. Yi, and S. Z. Li, "Online spatio-temporal structural context learning for visual tracking," in *Proceedings of the European Conference on Computer Vision*. Springer, 2012, pp. 716–729. 24

[173] B. Wu, H. Ai, C. Huang, and S. Lao, "Fast rotation invariant multi-view face detection based on real adaboost," in *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition*. IEEE, 2004, pp. 79–84. 25

[174] S. Wu, Y. Zhu, and Q. Zhang, "A new robust visual tracking algorithm based on transfer adaptive boosting," *Mathematical Methods in the Applied Sciences*, vol. 35, no. 17, pp. 2133–2140, 2012. 16

[175] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," 2015. 11

[176] ——, "Online object tracking: A benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2013, pp. 2411–2418. 41, 47

[177] Y. Xie, W. Zhang, C. Li, S. Lin, Y. Qu, and Y. Zhang, "Discriminative object tracking via sparse representation and online dictionary learning," *Cybernetics, IEEE Transactions on*, vol. 44, no. 4, pp. 539–553, 2014. 12

[178] B. Yang and R. Nevatia, "Online learned discriminative part-based appearance models for multi-human tracking," in *Proceedings of the European Conference on Computer Vision*. Springer, 2012, pp. 484–498. 11

[179] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, 2011. 3

[180] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 625–632. 13

[181] T. Yang and A. B. Chan, "Recurrent filter learning for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2010–2019. 23

[182] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel, "Robust tracking with weighted online structured learning," *Computer Vision–ECCV 2012*, pp. 158–172, 2012. 16

[183] X. You, Q. Li, D. Tao, W. Ou, and M. Gong, "Local metric learning for exemplar-based object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 8, pp. 1265–1276, 2014. 25, 68

[184] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 1349–1358. 22

[185] J. Zhang, S. Ma, and S. Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 188–203. 14

[186] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 127–141. 19

[187] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proceedings of the European Conference on Computer Vision*. Springer, 2012, pp. 864–877. 10, 39

[188] S. Zhang, H. Yao, X. Sun, and X. Lu, "Sparse coding based visual tracking: review and experimental comparison," *Pattern Recognition*, vol. 46, no. 7, pp. 1772–1788, 2013. 12

[189] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2042–2049. 38, 39

[190] ——, "Robust visual tracking via structured multi-task sparse learning," *International journal of computer vision*, vol. 101, no. 2, pp. 367–383, 2013. 14

[191] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja, "Robust visual tracking via exclusive context modeling," *Cybernetics, IEEE Transactions on*, vol. 46, no. 1, pp. 51–63, 2016. 14

[192] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4166–4174. 28

[193] L. Zhao, X. Gao, D. Tao, and X. Li, "Learning a tracking and estimation integrated graphical model for human pose tracking," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 26, no. 12, pp. 3176–3186, 2015. 14

[194] L. Zhaoping, "Theoretical understanding of the early visual processes by data compression and data selection," *Network: computation in neural systems*, vol. 17, no. 4, pp. 301–334, 2006. 5

[195] B. Zhong, Z. Chen, X. You, L. Li, Y. Xie, and S. Yu, "Robust weighted coarse-to-fine sparse tracking," in *IEEE International Conference on Security, Pattern Analysis, and Cybernetics*. IEEE, 2014, pp. 7–14. 12

[196] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proceedings of the IEEE Conference on Com-*

puter Vision and Pattern Recognition. IEEE, 2012, pp. 1838–1845. 11, 40

[197] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 943–951. 24

[198] Z. Zhu, G. Huang, W. Zou, D. Du, and C. Huang, "Uct: Learning unified convolutional networks for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1973–1982. 22

[199] Z. Zhu, W. Wu, W. Zou, and J. Yan, "End-to-end flow correlation tracking with spatial-temporal attention," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3, 21

[200] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 391–405. 25, 26, 68, 70, 86