Boosting Visual Object Tracking performance using a stack of Machine Learning algorithms

A Report submitted

in partial fulfillment for the award of the Degree of

BACHELOR OF TECHNOLOGY

in

AVIONICS

by

LITU ROUT

DEPARTMENT OF AVIONICS

INDIAN INSTITUTE OF SPACE SCIENCE AND TECHNOLOGY

THIRUVANANTHAPURAM - 695547, INDIA

MAY 2018

Boosting Visual Object Tracking performance using a stack of Machine Learning algorithms

A Report submitted

in partial fulfillment for the award of the Degree of

BACHELOR OF TECHNOLOGY

in

AVIONICS

by

LITU ROUT

DEPARTMENT OF AVIONICS

INDIAN INSTITUTE OF SPACE SCIENCE AND TECHNOLOGY

THIRUVANANTHAPURAM - 695547, INDIA

MAY 2018

This thesis is dedicated to my niece Shubhashree, nephews Aryan and Ayush, and my beloved Parents.

CERTIFICATE

This is to certify that the thesis entitled **"Boosting Visual Object Tracking performance using a stack of Machine Learning algorithms"** submitted by **Litu Rout**, to the Indian Institute of Space Science and Technology, Thiruvananthapuram, in partial fulfillment for the award of the degree of **Bachelor of Technology** in **AVIONICS**, is a bonafide record of the project research work carried out by him under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Deepak Mishra
Supervisor
Associate Professor
Department of Avionics
IIST, Thiruvananthapuram

Dr. R. K. Sai Subrahmanyam Gorthi
Co-Supervisor
Associate Professor
Department of Electrical Engineering
IIT, Tirupati

Dr. Manoj B. S.
Head of the Department
Department of Avionics
IIST, Thiruvananthapuram

Place: Thiruvananthapuram
MAY 2018

DECLARATION

I declare that this thesis titled **“Boosting Visual Object Tracking performance using a stack of Machine Learning algorithms”** submitted in partial fulfillment of the Degree of **“Bachelor of Technology”** is a record of original work carried out by me under the supervision of **Dr. Deepak Mishra** and **Dr. R. K. Sai Subrahmanyam Gorthi** and has not formed the basis for the award of any degree, diploma, fellowship, or other titles in this or any other Institution or University of higher learning. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

Litu Rout
Avionics
SC14B101

Place: Thiruvananthapuram
MAY 2018

ACKNOWLEDGEMENTS

I would like to express my thanks of gratitude to my supervisors **Dr. Deepak Mishra** and **Dr. R. K. Sai Subrahmanyam Gorthi** for giving me this opportunity to work on this project. I am very thankful to them for their consistent support, motivation and immense sharing of knowledge throughout my research.

I am grateful to **Dr. Manoj B. S**, Head of Department, Avionics, IIST for his support for the project. I am also thankful to all other faculties of Avionics Department for helping me to build my foundations strongly.

I am highly indebted to my family for extending their support during the ups and downs of my research. I thank the almighty for his blessing which had helped me to complete my work successfully.

Litu Rout

ABSTRACT

In the recent years, visual object tracking research has undergone significant improvement. The emergence of tracking by detection approach in tracking paradigm has been quite successful in many ways. Recently, deep Convolutional Neural Networks (CNN) have been extensively employed in most successful trackers. Yet, the standard approach has been based on correlation or feature selection with minimal consideration given to motion consistency. Thus, there is still a need to capture various physical constraints through motion consistency which will improve accuracy, robustness, and more importantly rotation adaptiveness. Therefore, one of the major aspects of this research is to investigate the outcome of rotation adaptiveness in visual object tracking. Also, our research includes various motion consistencies that turn out to be extremely effective in numerous challenging sequences with substantial improvement relative to deep learning based trackers: SiameseFC and CFNet.

Moving forward, we study Correlation Filter (CF) trackers, which are one of the most widely used categories in tracking. Though numerous tracking algorithms based on CFs are available today, most of them fail to efficiently detect the object in an unconstrained environment with dynamically changing object appearance. In order to tackle such challenges, the existing strategies often rely on a particular set of algorithms. Here, we propose a tracking framework that offers the provision to incorporate illumination and rotation invariance in the standard Discriminative Correlation Filter (DCF) formulation. We also supervise the detection stage of DCF trackers by eliminating false positives in the convolution response map. We further demonstrate the impact of displacement consistency on two widely appreciated CF trackers. The generality and efficiency of the proposed framework is illustrated by integrating our contributions into two state-of-the-art CF trackers: SRDCF and ECO. As per the comprehensive experiments on the VOT2016 dataset, our top trackers show substantial improvement of 14.7% and 6.41% in robustness, 11.4% and 5.27% in Average Expected Overlap (AEO) over the baseline SRDCF and ECO, respectively.

Later on, we study the efficacy of temporal regression with Tikhonov regularization in generic object tracking. Among other major aspects, we propose a different approach to regress in the temporal domain, based on weighted aggregation of distinctive visual features and feature prioritization with entropy estimation in a recursive fashion. We provide a statistics based ensembler approach for integrating the conventionally driven spatial regression results (such as from ECO), and the proposed temporal regression results to accomplish better tracking. Further, we exploit the obligatory dependency of deep architectures on provided visual information, and present an image enhancement filter that helps to boost the performance on VOT2016 dataset. Our extensive experimentation shows that the proposed weighted aggregation with enhancement filter (WAEF) tracker outperforms the baseline (ECO) in almost all the challenging categories on popular OTB50 dataset with a cumulative gain of 14.8%. As per the VOT2016 evaluation, the proposed framework offers substantial improvement of 19.04% in occlusion, 27.66% in illumination change, 33.33% in empty, 10% in size change, and 5.28% in average expected overlap.

Contents

List of Tables

List of Figures

Abbreviations

bbox	Bounding Box
HOG	Histogram of Oriented Gradients
CNN	Convolutional Neural Network
KCF	Kernelized Correlation Filters
SRDCF	Spatially Regularized Discriminative Correlation Filters
VOT	Visual Object Tracking
OTB	Object Tracking Benchmark
RI	Rotation Invariance
CFNet	Correlation Filter Network
SiameseFC	Siamese Fully Convolutional
IC	Illumination Correction
RA	Rotation Adaptive
ECO	Efficient Convolution Opearators
CF	Correlation Filter
WA	Weighted Aggregation
EF	Enhancement Filter
TR	Temporal Regression
FPEE	Feature Prioritization with Entropy Estimation
WAEF	Weighted Aggregation with Enhancement Filter

Chapter 1

Introduction

Computer vision research started in the early 1960s as an artificial intelligence problem, and the goal was to help machines perceive the world through images. The research work in computer vision based analysis and understanding the motion of a single object in a sequence of images began around 1970s. David Marr, based on his background in mathematics and neuroscience, proposed the theory of computer vision [1] which stated that a complete geometric reconstruction of the observed scene is required for visual perception. According to him, a vision system should be able to estimate the shape, color, orientation etc. of various objects present in the acquired visual inputs, such as images and videos of a scene. In 1990s researches started migrating from Marr's theory to another theory called as active vision theory [2, 3] which actively changes the sensor orientation and location to obtain additional informations. Active vision deosn't require a complete representation of the environment, instead it defines individual visual tracks which operates independently to perform visual perception. Although active vision related research grew highly during early 90s, using active sensors for visual perception was not practically feasible.

In late 1990s, due to the advancement of various high speed computers, various motion-based approaches were developed to solve the computer vision tasks. The objective of the motion-based approaches was to interpret the sequence of frames (or images) from videos. Working on a sequence of images added an additional dimension: time and a new constraint: temporal coherence. Addition of these factors had given a remarkable development in the research of video processing. One of the important development was the complete surveillance system which consists of the following components:

- **Object detector**, which detects the region of interest.
- **Object tracker**, which estimates the trajectory of the object.
- **Object classifier**, which classifies the desired object from various classes.

- **Activity recognizer**, which analyses the activities and behaviours of the tracked objects.

Additionally, visual object tracking finds applications in diverse fields like traffic monitoring, surveillance systems, human computer interaction etc. Though the same object is being tracked throughout a given video sequence, the conditions under which the video is captured may vary due to changes in the environment, object, or camera. Illumination variations, occlusion, motion blur, object deformations, object rotations etc. are various challenges that occur due to changes in aforementioned factors. A good tracking algorithm should continue tracking a desired object and its performance should remain unaffected under all these conditions.

Estimating similarity across various image patches is one of the most fundamental components in the field of object tracking. In most of the cases, a precise similarity measure leads a solid foundation for several challenging tasks which include structure from motion, wide baseline matching, object recognition, segmentation, classification and image retrieval [4]. Although great achievements have been accomplished within the last few years, there is still a need to optimize the performance in terms of accuracy, robustness, and speed.

Over the years, many tracking algorithms have been proposed [5] where the main goal is to localize an object in a series of frames with the sole supervision of a bounding box given only in the first frame of the sequence. In these scenarios the object's appearance model is learnt from previous frame of a sequence online and these learnt models are then cross-correlated with the search image to localize the target object in the next frame. There are several state-of-the-art methods based on similarity rationale with certain modifications such as KCF [6], SRDCF [7] and C-COT [8] which are widely accepted by the tracking community. Due to the tremendous achievements of deep neural networks in diverse computer vision applications, the researchers have focused their attention to bring the best out of deep convolutional nets in tracking paradigm. Moreover, the scarcity of large sets of supervised data and extremely slow learning ability have made deep trackers not feasible for real life applications [9]. Even though there is a massive constraint of speed, deep trackers such as DeepSRDCF [10], TCNN [11] and MDNet [12] have proved their effectiveness in wide variety of challenging sequences.

Some of the recent trackers have aimed at learning a detector per video by fine-tuning multiple layers of a pre-trained deep network with stochastic gradient descent mechanism [12, 13]. But the necessity of high frames per second in real world applications leaves the online adaptive deep convolutional networks a step behind the other state-of-the-art trackers [5, 14]. A possible solution to these shortcomings could be to use a pre-trained deep similarity network such as Siamese network [15] in order to discriminate the target from its background. So the objective of the network would be to learn from a single exemplar image in one branch and predict the essential parameters of the other branch which will assist in identifying instances of the same object in the upcoming frames [15]. Thus, the deep network would generalize a similarity function from annotated pairs of raw image patches without attempting to use hand crafted features [16].

Most of the trackers have achieved appealing results both in accuracy and robustness. It is also witnessed that the use of several consistency techniques such as scale adaptive KCF [17], influence of windowing [9], bounding box regression [12] and online adaptation of appearance model [12] has turned out to be extremely effective in numerous tough sequences. Even though the introduction of immunity to minor scale changes has brought radical advancement, still there is necessity of orienting the bounding box according to the target object. In real life applications, detecting not only proper bounding boxes but also estimating the orientation of the target object plays a vital role. This would be a key factor in increasing overlap ratio and anticipating trajectory of the target object in a more efficient manner. One of the major advantages of using orientation would be to update the appearance model and to use more sophisticated features from the rotated version of cropped exemplar image in the subsequent frames.

Therefore, one of the important aspects of this research is to propose a scheme to determine the orientation of the target object and analyse its impact in visual object tracking. We further extend our work and propose a better approach of updating target position taking into account the distance and direction of motion and Gaussian weighted average response map based on various scales. Our proposed method is generic and it can be integrated with other state-of-the-art trackers which results in substantial improvement in the performance standard. To establish this proposition, we have integrated our algorithm with SiameseFC [9] and CFnet [18]. The main reason behind choosing these two trackers is that they achieve comparable state-of-the-art performance while operating at extremely high frames per second. The proposed algorithm have been evaluated on popular tracking benchmarks such as VOT [5] and OTB [14]. After successfully experimenting our ideas, namely rotation adaptiveness and motion consistencies on the two aforementioned deep learning based trackers, we aim at proving the generality of the proposed schemes by integrating these contributions into the standard discriminative correlation filters, which is one of the core components in visual object tracking. In contrast to the aforementioned rotation adaptive deep learning based models, this method incoporates rotation adaptiveness directly in the standard discriminative correlation filters. In the following few paragraphs, we briefly describe the essence of discriminative correlation filter trackers in generic object tracking paradigm.

Most of the existing trackers can be classified as either generative or discriminative models. The generative trackers [19, 20, 21, 22, 23] use the object information alone to search for the most probable region in an image that matches the initially specified target object. On the other hand, the discriminative trackers [24, 25, 26, 27, 28] use both the object and background information to learn a classifier that discriminates the object from its background. The discriminative trackers, to a large extent, make use of Correlation Filters (CF) as classifiers. The main advantage of CFs is that correlation can be efficiently performed in the Fourier domain as simple multiplication, as proven by Parseval's theorem. For this reason, CF trackers are learned and all computations are performed efficiently in the Fourier domain with drastic reduction in computational complexity [26]. Thus, the CF trackers have gained popularity in the community because of their strong discriminative power, which emerges due to implicit inclusion of large number of negative samples in training.

Despite all the advancements in CF tracking, most of these CF algorithms are still not robust enough to various tracking challenges. These limitations are due to the inherent scarcity of robust training features that can be derived from the preceding frames. This restricts the ability of the learned appearance model to adapt the changes in target object. Therefore, we propose rotation adaptiveness and illumination correction schemes in order to extract sophisticated features from previous frames that helps in learning robust appearance model and filter parameters. The rotation adaptiveness, up to some extent, tackles the issues of object deformation due to the robustness in representation.

In spite of the effort devoted by a large part of the community, there are still several challenges yet to be conquered. To overcome such challenges, most of the previously proposed trackers focus on some of the key components in tracking, including robust feature extraction for learning better representation [29, 30, 31, 32], accurate scale estimation [33], rotation adaptiveness [34, 35], motion models [36] etc. There are several other state-of-the-art trackers such as SRDCF [7], and CCOT [8] that implement additional constraint on the residual sum of errors to enforce higher degree of smoothness on the physical movement of the object. In the pursuit of accurate tracking, some of the proposed frameworks [32, 37] are predominantly attributed by sophisticated features and complex models. Further, the emergence of deep CNN has replaced the low-level hand-crafted features which are not robust enough to discriminate significant appearance changes. The success of deep learning based trackers such as MDNet [12] and TCNN [11] on popular tracking benchmarks such as OTB [14] and VOT [5] is a clear indication of the distinctive feature extraction ability of deep CNN. In spite of the popularity, these feature extractors still lack high quality visual inputs that can further boost the performance. Therefore, one of the major aspects of this thesis is to study the effect of enhancing visual inputs prior to feature extraction. In some sequences like Matrix (Figure 1.1), the hand-crafted and CNN features, as used in ECO, also fail to track the target, whereas image enhancement leads to sophisticated feature extraction that helps in tracking under such conditions.

Though deep learning based models have gained a lot of attention on account of their accuracy and robustness, the inherent scarcity of data, and required time for training these networks online, leave such models a step behind the correlation filter (CF) trackers. For this reason, a proper synthesis of CNN as feature extractor, and CF as detector has been doing exceedingly well in most of the challenging sequences. However, most of these fusion based trackers[8, 38], being supervised regressors, learns to maximize the spatial correlation between target and candidate image patches. Due to spatial regularization, as in SRDCF [7], such trackers are capable of searching in a large spatial region that pro-

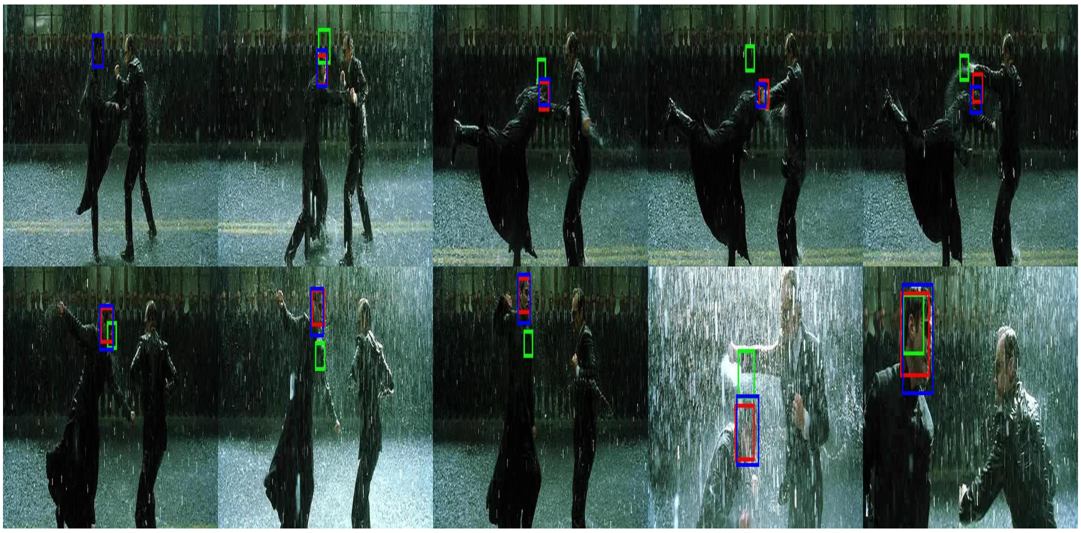

Figure 1.1: The groundtruth of Matrix sequence of VOT2016 is shown in blue. The ECO (green) tracker fails to track the object because of drastic appearance changes. However, our ECO_EF (red) can handle the abrupt transition in appearance, mainly due to the enhanced visual information provided before feature extraction, and tracks successfully.

duces a significant gain in performance. But, these classifiers give minimal consideration to regress in temporal domain. Therefore, we exploit the temporal regression (TR) ability of a simple, yet effective model considering weighted aggregation of preceding features.

1.1 Objective of the Thesis

One of the major aspects of this thesis is to study comprehensively visual object tracking in the real world scenarios. We critically analyze various state-of-the-art trackers, and identify their shortcomings. Thereafter, we intend to augment these state-of-the-art trackers with our own contributions to tackle the identified issues efficiently. Thus, we aim at contributing in the expansion of tracking research with extenssive experimentation and compelling inferences. In the process of our research, we have identified few issues in the current state-of-the-art trackers and contributed in those areas focusing on relative gain in overall performance. Our contributions are summarized below.

1.2 Contributions of the Thesis

With respect to deep learning based trackers: SiameseFC [9] and CFNet [18], our contributions are summarized as following:

- At the beginning, we study the existing centroid update strategy, and propose an aproach to enhance the degree of smoothness on physically varying movement variables, such as speed and angular displacement (Section 3.2.3).

- Further, we propose to exploit the scale invariant features in determining the object's location, which is based on Gaussian weighted average response of the scale pyramid (Section 3.2.4).

- Thereafter, we explore the rotation adaptiveness in Siamese architecture, and propose to augment various rotations in determining the centroid location as well as orientation efficiently in the next frame (Section 3.2.5).

The major contributions of our research with correlation filter based trackers: SRDCF [7] and ECO [38] are as follows:

- An Illumination Correction filter (IC) (Section 3.3.1) is introduced in the tracking framework that eliminates the adverse effects of variable illuminations on feature extraction.

- We propose an approach to incorporate rotation adaptiveness (Section 3.3.3) in standard DCF by eliminating false positives (Section 3.3.3.3) and optimizing the orientation (Section 3.3.3.4) of the target object in the detector stage. The orientation optimization helps in extracting robust features from properly oriented bounding boxes unlike most state-of-the-art trackers that rely on axis aligned bounding boxes.

- Building on it, we supervise the sub-grid localization cost function (Section 3.3.3.5) in the detector stage of DCF trackers. This cost function is intended to maximize the ratio of response score and euclidean distance between target centroids from immediate past frame and test frame.

- Further, we show the impact of imposing higher degree of smoothness (Section 3.3.4) on two popular CF trackers, namely Spatially Regularized Disriminative Correlation Filters (SRDCF) [7], and Efficient Convolution Operators (ECO) [38].

The proposed technical and theoretical contributions with respect to Temporal Regression (TR) are summarized as following:

- A simple and effective enhancement filter (EF) (Section 3.4.2) is proposed to alleviate the adverse conditions in visual inputs prior to feature extraction. By this approach, the proposed tracker is able to perform against the state-of-the-art on VOT2016 dataset with an improvement of 5.2% in Average Expected Overlap (AEO) over the baseline approach.

- Although a lot of methods have been developed based on spatial regression, TR still remains a relatively less explored method in tracking. Therefore, in this thesis, a detailed analysis on impacts of employing TR in single object tracking is undertaken (Section 3.4.3).

- For efficient learning of TR parameters, a weighted aggregation (Section 3.4.3.1) based approach is proposed to suppress the dominance of un-correlated frames while regressing in temporal domain. Also, the training features are further organised based on average information content (Section 3.4.3.2). To our knowledge, this is in contrast to the conventional linear regressions in which equal [6], or more preference [18] is given to the historic frames. In order to generalize better, and control over-fitting in temporal domain, we have embedded the whole TR framework in Tikhonov regularization (Section 3.4.3.3).

Though we have demonstrated the importance of contributions through integrating with SiameseFC, CFNet, SRDCF, and ECO, the proposed framework is generic, and can be well integrated with other trackers to tackle some of the aforementioned tracking challenges with certain improvement in accuracy.

1.3 Thesis Overview

In this thesis, we have consolidated our contributions in various state-of-the-art trackers aiming at advancement in generic object tracking research. The thesis is structured in a sequential approach based on our contributions. At first we explore the weaknesses of top performing trackers on popular benchmarks, and analyze the efficacy of our contributions in those models. After verifying that the proposed contributions perform favourably on numerous challenging sequences, we incorporate these contributions in the standard discriminative correlation filter based trackers, which are the widely appreciated tracking frameworks in the vision community. Thus, we propose a generic framework to eradicate several tracking issues to a great extent. Thereafter, we augment temporal correspondence with these spatial-domain correlation filter trackers in order to boost the overall tracking performance. For unbiased assessment of our trackers, we have used toolkits provided by various tracking benchmarks for comprehensive evaluation, and draw compelling inferences from the obtained results. To assess the genericness of the proposed contributions, we evaluate the trackers on diverse benchmarks, and show considerable gain in the corresponding evaluation metrics.

Chapter 2

Visual Object Tracking

2.1 Object Tracking : State-of-the-art

Object tracking helps us to understand and describe the object behaviour by replacing the traditional method of monitoring computer by human operators. Every tracking algorithm requires an object detection mechanism either in every frame or when the object first appears in the video. Numerous approaches have been proposed for tracking, however, approaches differ from each other due to various factors such as:

- which object representation is suitable for tracking
- which image feature should be used for tracking
- how to model the motion, appearance and shape of the object

The solution to the above problems depends on the environment in which tracking is performed and also application of the tracking.

2.1.1 Representation of Object in Tracking

Inorder to track any object, it can be represented in various forms as discussed below.

- **Point:** The object can be represented using a single point e.g., centroid as shown in Figure 2.1(a). This is used mainly when the object to be tracked occupies very small region in the image.
- **Primitive geometric shapes:** The object can also be represented using a rectangle as in Figure 2.1(b) or an ellipse as in Figure 2.1(c). These are mostly suitable for representing simple rigid objects however, it can be used for tracking a variety of non-rigid objects also.

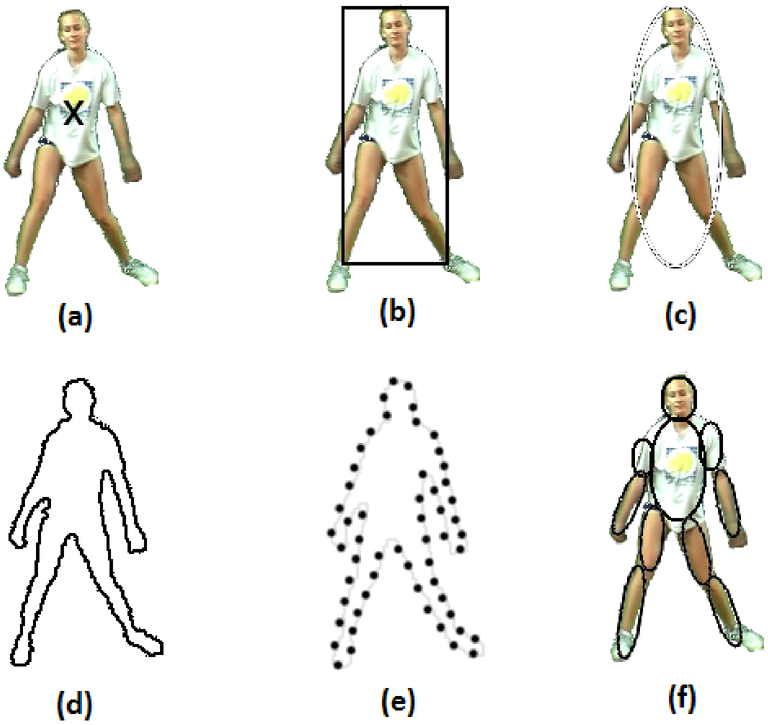

Figure 2.1: Object representations [39] : (a) Centroid (b) Rectangular (c) Elliptical (d) Object contour (e) Control points on the contour

- **Object contour:** The contour represents the boundary of the objects and are used for representing non-rigid objects as shown in Figure 2.1(d,e). These are mainly used for complex non-rigid shapes. The region inside the contour is called as silhouettes.

- **Articulated shape models:** Articulated object refers to the object parts that are held with joints. Human body is an articulated object with legs, hands, head, torso and feet fully connected by joints. The relation between these parts are governed by different model parameters like joint angle. To represent each part of the articulated model, we can use lines, ellipse or cylinders as shown in Figure 2.1(f) . Some of the articulated object tracking techniques are discussed in [40], [41], [42].

Based on the appearance, some of the object representations are as below,

- **Probability densities of object appearance:** The probability density of the object can be a gaussian model, gaussian mixture model, or non-parametric like parzen windows or histograms.

- **Templates:** Template model can be created using the basic shape of geometry or object silhouettes. The disadvantage of template model is that it considers only one view of the object and hence it is only suitable for tracking objects whose pose doesn't vary.

- **Multi-view appearance model:** It considers different views of the object by a subspace decomposition like Eigenspace decomposition or by training a classifier like Support Vector Machine. The major disadvantage of this is that the appearance information of all view of the object should be known priorly.

2.1.2 Feature Selection for Tracking

All tracking algorithms require a set of unique features to represent the object. Selection of features play an important role in the tracking process because few of the features are not stable with various tracking challenges like occlusion, scale, illumination variations etc. Feature selection also depends on object representation i.e., for histogram based appearance, color is used as the feature whereas for contour based representations object edges are usually used as the feature. However, some of the tracking algorithms uses combination of various features to improve its performances. Few of the visual features are as discussed below,

- **Color:** It mainly depends on two factors, (i) spectral power distribution of the illuminant and (ii) surface reflectance property of the object. Though color spaces are highly sensitive to noises, it is one of the mostly used features.

- **Edges:** Object boundaries have high difference in image intensities and hence are another important feature for tracking. It is independent of illumination variation and hence is a better option as compared to that of color features.

- **Optical Flow:** It defines the displacement vector corresponding to the translation of each pixel in an image. It is mostly used as a feature in motion based segmentation and tracking applications.

- **Texture:** It refers to the intensity variation of a surface which defines properties like smoothness and regularity. Unlike the color feature which is obtained directly from the image, texture feature requires a processing step to generate the descriptor to define the target.

Mostly the selection of features depend on the application of the tracking. Other commonly used features include Histogram of Oriented Gradients (HOG), color-name etc.

2.1.3 Challenges in Tracking

Detection of object can be a challenging task as the object can have complicated structure, or it may change in size, shape, location and orientation over the subsequent frames. Currently a large number of tracking algorithms are available, however most of these may contain error that will drift the object of interest. Better the tracking algorithm, lesser will be the drift. Few of the major tracking challenges include:

- **Clutter:** During tracking of the object, sometimes the background may be cluttered or may be surrounded by other objects as shown in Figure 2.2. This make estimation of target a difficult task.

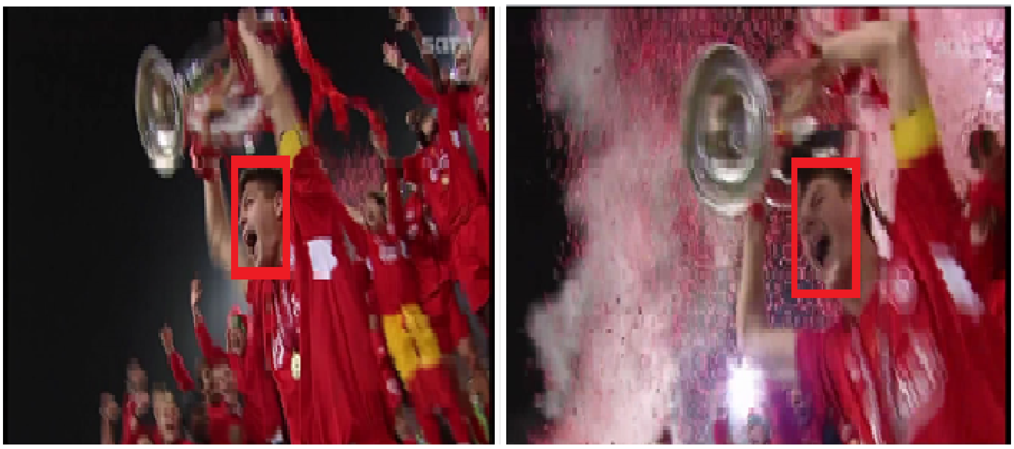

Figure 2.2: Background clutter

- **Illumination Variation:** The object to be tracked may be exposed to various background illuminations as shown in Figure 2.3. By keeping the local background along with the model, we can overcome this challenge.

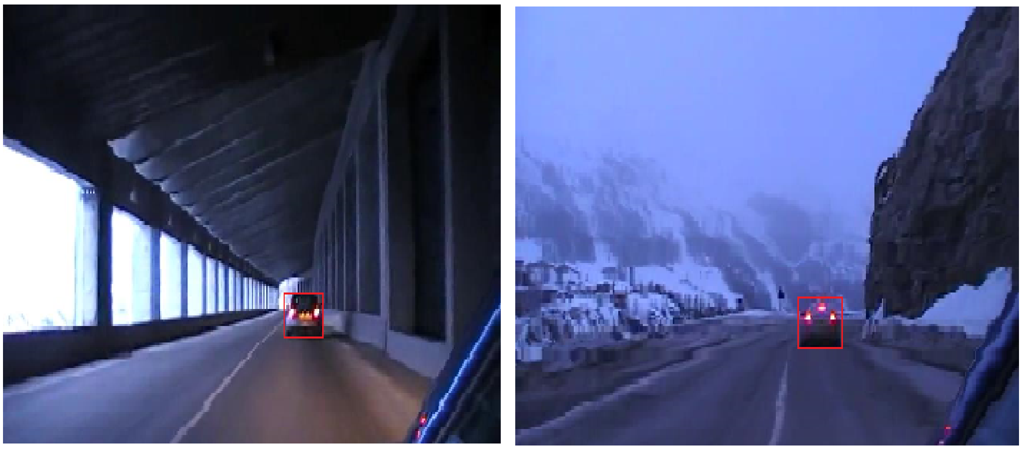

Figure 2.3: Illumination variation

- **Occlusion:** In a video, if the target object falls behind another object in the current image, then the target is said to be occluded as shown in Figure 2.4.

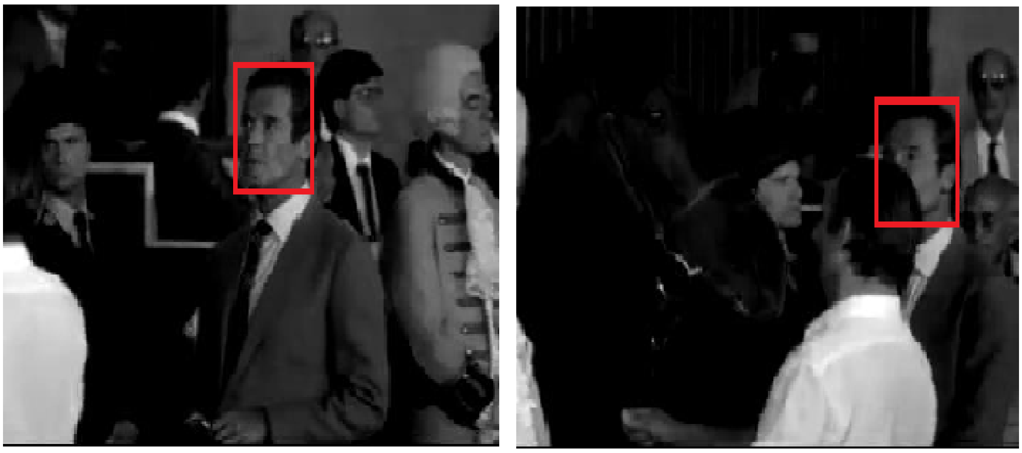

Figure 2.4: Occlusion

- **Scale Variation:** Size of the object changes due to the zooming in and out of camera as illustrated in Figure 2.5.
- **Shape:** Shape of the object may vary along the video as shown in Figure 2.6.

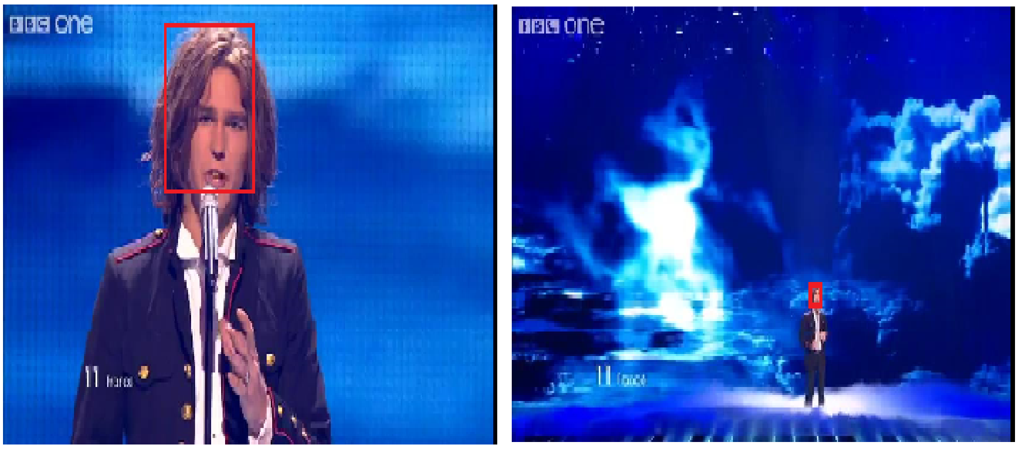

Figure 2.5: Scale Change

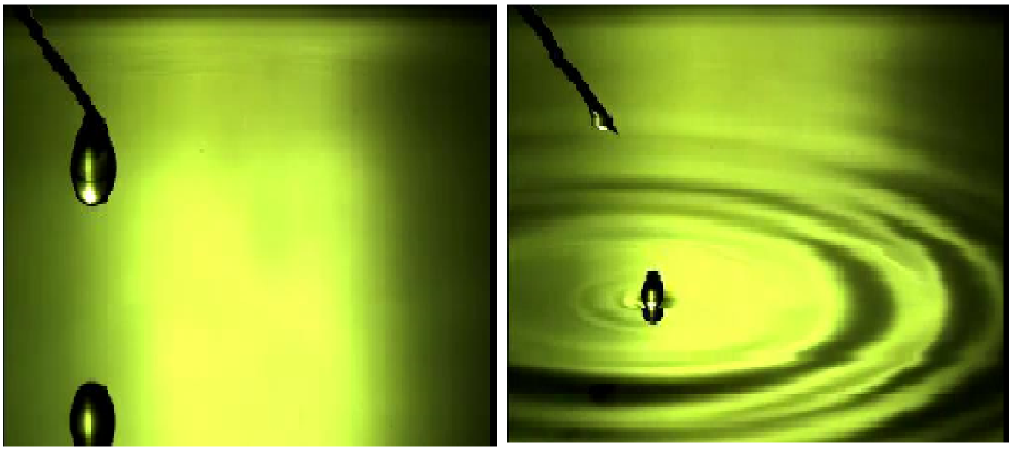

Figure 2.6: Shape

- **Motion Blur:** Due to motion of the target or camera, object may appear to be blurred as shown in Figure 2.7. The appearance model of the tracking algorithm may get highly affected due to this tracking challenge.

2.1.4 Tracking Applications

- **Automated Surveillance:** It is a task monitoring scene to capture the suspicious activities as show in Figure 2.8. It can be used to acquire obstacle avoidance capabilities in case of robot navigation.

- **Human Computer Interaction (HCI):** It is used as a gesture recognition, eye gaze tracking for data inputs to computers as shown in Figure 2.9. Using a robust tracker,

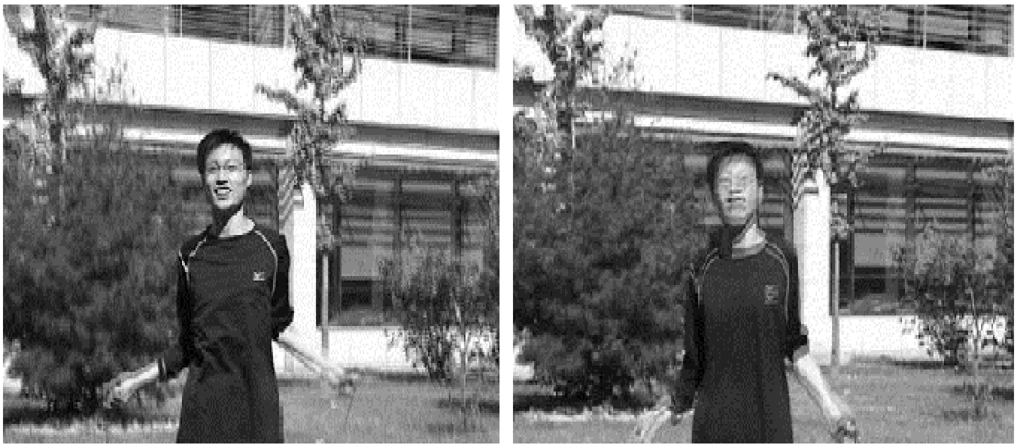

Figure 2.7: Motion Blur

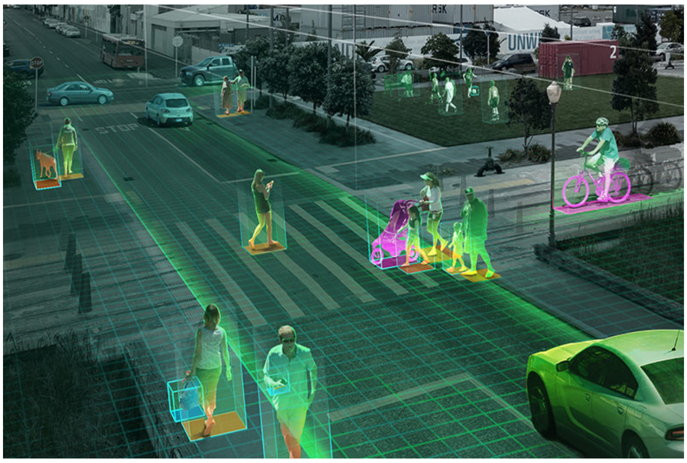

Figure 2.8: Automated Surveillance

the user can interact with the system based on gestures.

Figure 2.9: Human Computer Interaction (HCI)

- **Traffic Monitoring:** It involves real-time capturing of various details like number plates of vehicle passing, speed at which they are travelling, with the help of which one can direct the traffic flow as shown in Figure 2.10.

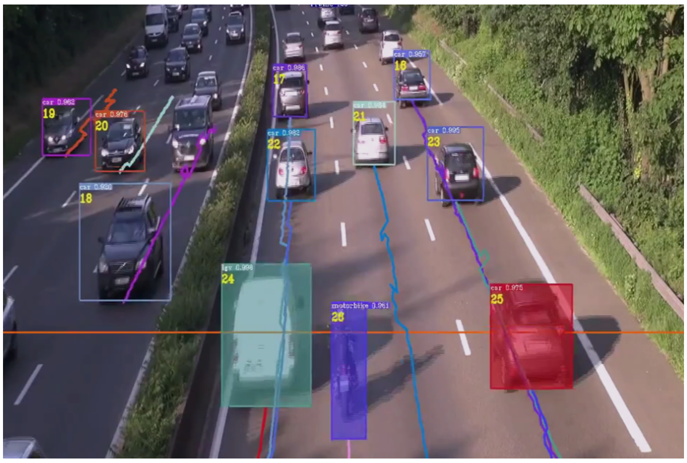

Figure 2.10: Traffic Monitoring

- **Autonomous Vehicles:** In autonomous vehicle design, as shown in Figure 2.11, some of key components are detection, segementation, and trajectory estimation of several objects nearby. The emergence of high commercial demands has played an important role in the rapid progress of visual object tracking.

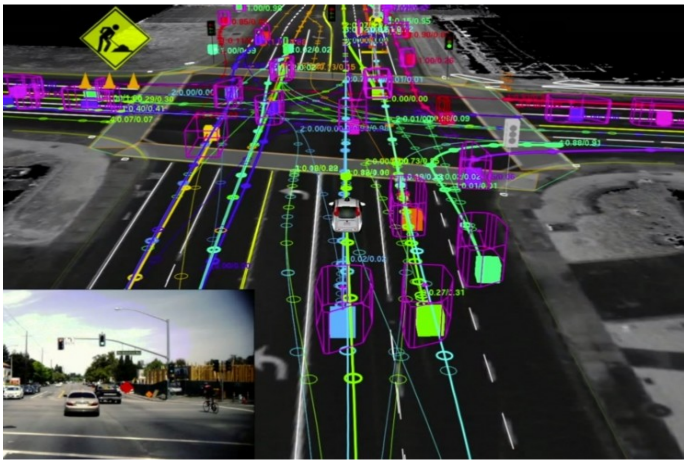

Figure 2.11: View of a driverless car

2.2 Literature Review

Most of the correlation filter based algorithms are based on two key elements, how the target object is represented and how to localize this object of interest in the subsequent frames. Object representation models have developed gradually from histogram [43] based approach to more advanced generative [44, 45] or discriminative [6, 46] approaches. For target object localization, methods such as Elliptical head tracking [47], Probabilistic color and adaptive multi-feature tracking [48], Robust visual tracking [49] and Learning to track with multiple observers [50] have gained a lot of attention. Recently, the widespread success of object detection algorithm has emerged an advanced approach of localization, known as tracking-by-detection. Due to outstanding performance of these tracking-by-detection algorithms on evaluation benchmarks [5, 14], this paradigm has gained popularity in the tracking community. This method usually employ binary classifier to discriminate target object from its background. S. Hare *et al.* discuss Struck [51], a discriminative tracker which employs a kernelized structured output Support Vector Machine(SVM) to provide adaptive tracking. M. Danelljan *et al.* have proposed SRDCF [7] which uses a spatially regularized correlation filter that helps in learning from a large set of negative samples, without corrupting the positive samples. Y. Li *et al.* have proposed a scale adaptive scheme in [17] which has strong impact on determining the size of the target efficiently. G. Koch *et al.* explores a method to train Siamese neural network which ranks the similarity between its input image patches [52]. J. Valmadre *et al.* take a step forward to investigate the influence of modified Alex-net on Siamese network [9] and propose an end-end training model of correlation filter in CFnet [18].

Sequence like glove [5] from VOT challenge as shown in Figure 2.12 undergoes severe deformation which leads to significant changes in aspect ratio as well as rotation of the bounding box. Due to this, the ill-equipped axis aligned bounding box used in most state-of-the-art tracking methods fails to capture more detailed information from the object of interest. Y. Hua *et al.* [53] have addressed this issue by generating suitable candidates which capture more detailed information by estimating the transformations undergone by the object. H. A. Rowley *et al.* have carried out an extensive research in designing a rotation invariant neural network based face detection system [54]. There are several face

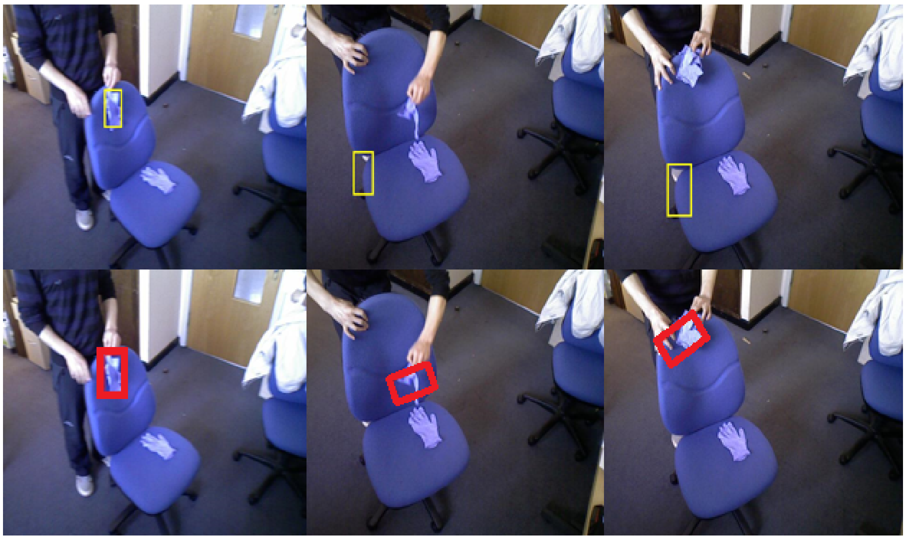

Figure 2.12: Sample frames from glove sequence regarded as one of the toughest sequences according to VOT 2016 results [5]. First column indicates the ground truth bounding box in the first frame. Our modified SiameseFC(red) successfully tracks the geometric deformations unlike original SiameseFC[9](yellow).

detection systems which could only detect upright or frontal faces [55]. But this is far from the reality where images of faces could be much more complicated than just upright or frontal. Introduction of router network in [54] to detect angle of orientation of the face has helped to convert the rotated faces back to frontal before it is passed through the face detector. M. Jaderberg *et al.* have introduced a spatial transformer network (STN) [56], where the model learns invariance to rotation, scale, translation and other warping of the input data. Use of localization network, sampling grid generator and sampler in a STN have brought radical achievement in challenging object classification datasets such as CUB-200-2011 birds dataset [57]. In this thesis, we explore extensively the impact of rotation invariance in tracking paradigm as well as several new consistency techniques which have outperformed the original tracking algorithm by a large margin.

Numerous variants of the basic CF tracker have been proposed by adding constraints to the basic filter design and by utilizing different feature representations of the target object. Initial extensions start with the KCF tracker [26] which uses a kernel trick to perform efficient computations in the Fourier domain. The Structural CF tracker [27] uses a part based technique in which each part of the object is independently tracked using separate CFs. Danelljan *et al.* [28] proposed the SRDCF tracker which uses a spatial regularizer to weigh the CF coefficients in order to emphasize the target locations and suppress the background information. Thus, the SRDCF tracker includes a larger set of negative patches in training, leading to a much better discriminative model.

The earlier trackers directly used the image intensities to represent the target object. Later on, feature representations such as color transformations [58, 59, 60, 26], Color-names [37] etc. were used in the basic CF trackers. Due to the significant advancement of deep neural networks in object detection and recognition tasks, features from these networks have also found applications in visual tracking, giving rise to substantial improvement in performance. The deep trackers, such as DeepSRDCF [10], MDNet [12], and TCNN [11], clearly indicates the distinctive feature extraction ability of deep networks. The HCF tracker [61] exploits both semantic and fine-grained details learned from a pre-trained Convolutional Neural Network (CNN). It uses a multi-level correlation map to locate the target object. The CCOT tracker [8] uses DeepSRDCF [10] as the baseline tracker and incorporates an interpolation technique to learn the filter in continuous domain with multi-resolution feature maps. The ECO tracker [38] reduces the computational costs of CCOT by using a factorized convolution operator that acts as a dimensionality reduction operator. ECO also updates the features and filters after a predefined number of frames, instead of updating after each frame. This eliminates redundancy and over-fitting to recently observed samples. As a result, the deep feature based ECO tracker does reasonably well on diverse datasets outperforming the other CF trackers by a large margin.

Among rotation adaptive tracking, Zhang *et al.* [62] propose an exhaustive template search in joint scale and spatial space to determine the target location, and learn a rotation template by transforming the training samples to Log-Polar domain. We learn rotation adaptive filter in the cartesian domain by incorporating orientation in the standard DCF. In contrast to a recent rotation adaptive scheme, as proposed by Rout *et al.* [35], we incorporate rotation adaptiveness directly in the standard DCF formulation, by performing a pseudo optimization on a coarse grid in the orientation space. Qianyun *et al.* [63] use a multi-oriented Circulant Structure with Kernel (CSK) tracker to get multiple translation models each dominating one orientation. Each translation model is built upon the KCF tracker. The model with highest response is picked to estimate the object's location. The main difference is that we do not learn multiple translation models at various orientations, as proposed in multi-oriented CSK. In contrast, we optimize the total energy content in convolution responses at the detector stage with respect to object's orientation. The multi-channel correlation filter is then learned from a set of training samples which are properly oriented in a deterministic approach. Note that, our training process requires a single model.

There are many recently proposed tracking algorithms that offer superior performance on the VOT2017 public as well as sequestered dataset [5]. The LSART tracker [64] complements the kernelized ridge regression technique with CNN. It also combines the deep features with handcrafted features. The CFWCR tracker [65] extends the ECO tracker based on a continuous convolution operator and uses weighted convolution responses of the CNN features. CSR-DCF [66] improves the DCF trackers using spatial and channel reliability concepts using HOG and colornames as features.

Correlation Filter (CF) based trackers have gained a lot of attention due to their low computational cost, high accuracy, and robustness. The regression of circularly shifted input features with a Gaussian kernel makes it plausible for implementation in Fourier domain, which in fact is the predominant cause of low computational cost. The object representation models, as adapted by many such trackers, have emerged gradually with colour attributes [31], HOG [67], SIFT [32], sparse based[29], CNN [8], and hierarchical CNN [61]. These methods have assisted in diminishing the adverse effects of ill-posed visual inputs. As discussed in this thesis, our proposed enhancement filter, in a loose sense, contributes towards alleviating this issue further by pre-processing the inputs prior to feature extraction.

Among spatio-temporal models, the Spatio-Temporal context model based Tracker (STT) [68] proposes a temporal appearance model that captures historical appearances to prevent the tracker from drifting into the background. Also, STT proposes a spatial appearance model that creates a supporting field which gives much more information than the appearance of the target, and thus, ensures robust tracking. The Recurrently Target-attending Tracker (RTT) [69] exploits the essential components of the target in the long-range contextual cues with the help of a Recurrent Neural Network (RNN). The close form solution used in RTT is computationally less intensive, and more importantly, it helps in mitigating occlusion cases upto a great extent. The deep architecture proposed in [70] consists of three networks: a Feature Net, a Temporal Net, and a Spatial Net which assist in learning better representation model, establishing temporal correspondence, and refining the tracking state, respectively. The Context Tracker [71] explores the context on-the-fly by a sequential randomized forest, an online template based appearance model, and local features. The distracters and supporters, as proposed in Context Tracker, are very much useful in verifying genuine targets in case of resumption. The TRIC-track [72] algorithm uses incrementally learned cascaded regression to directly predict the displacement between local image patches and part locations. The Local Evidence Aggregation [73], as per the discussion in TRIC-track, determines the confidence level which is used to update the model. The Recurrent YOLO (ROLO) [74] tracker studies the regression ability of RNN in temporal domain.

In a nutshell, most of the trackers try to incorporate temporal information through complicated learning strategies often leading to low performance and high time complexity. On the contrary, we intend to use a simple, yet effective weighted aggregation and feature prioritization strategy while regressing in the temporal domain. Our comprehensive experimentation indicates that the closed-form solution of TR, as proposed in this thesis, can capture temporal correspondence very effectively without hampering the time complexity much.

Chapter 3

Proposed Methodology

3.1 Introduction

Here, we elaborate our contributions with appropritate justification of each individual component. We begin our description with the contributions made in deep learning based trackers (Section 3.2), followed by discriminatative correlation filters (Section 3.3), and finally, weighted aggregation with enhancement filter (Section 3.4).

3.2 Contributions in Deep Learning based Trackers

In this section, we detail our contributions by proposing a generic approach for incorporating rotation invariance (RI) in object tracking and introducing the motion consistencies guided by the laws governing physical motion of the objects. For the sake of experimentation and analysis we incorporated the proposed modifications to Siamese Net and CFnet. Here we first briefly discuss about the architecture of SiameseFC in Section 3.2.1 and CFnet in Section 3.2.2 followed by the consistencies termed as Displacement Consistency in Section 3.2.3, Scale Consistency in Section 3.2.4, and RI in Section 3.2.5.

3.2.1 Siamese Fully Convolutional Network

Deep similarity learner mainly learns the parameters of a function $f(z, x)$ which takes two images as input and generates a response score. If the two images depict the same object, it generates a high score otherwise a low score. A convolutional neural net is used as this learning function $f(z, x)$. Typically Siamese architectures have been quite successful in deep similarity measure. The network learns the parameters from the first frame of each sequence and then all the possible candidates are tested exhaustively to measure similarity with the exemplar image. A simple architecture of Siamese fully convolutional network is shown in Figure 3.1. In this architecture, the appearance model of the exemplar image isn't updated at all. Siamese network employs same transformation ϕ which is a five layered CNN to both of its inputs z and x. The transformed inputs $\phi(z)$ and $\phi(x)$ are cross-correlated to obtain a response map. The location of maximum response score indicates the position of the object of interest. Thus, the overall similarity function becomes $f(z, x) = g(\phi(z), \phi(x))$. One of the major advantages of using fully convolutional architecture is that, a larger search region can be fed into the network without resizing it to the size of exemplar. Then the similarity function helps in finding the response score at all translated sub-windows in a large search region. The response map computed by this architecture is given by equation (3.1).

$$f(z, x) = \phi(z) * \phi(x) + b \tag{3.1}$$

where b ϵ **R** is a bias signal.

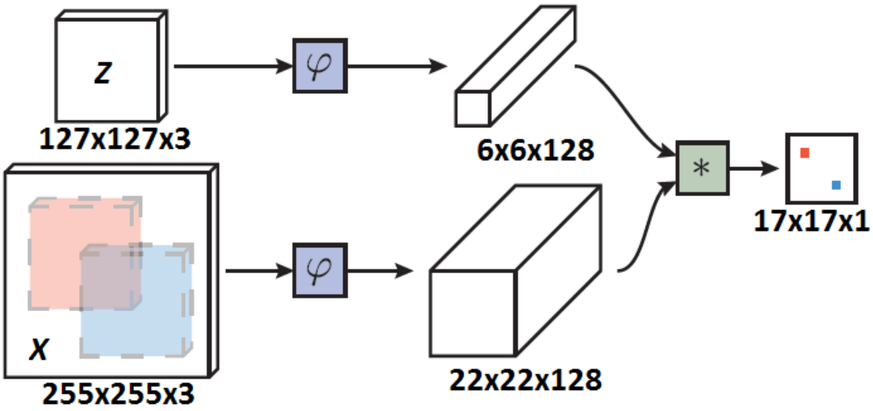

Figure 3.1: Siamese Fully Convolutional Network. z-branch and X-branch are known as exemplar and instance branch respectively. Z is the target exemplar image extracted from the first frame. Siamese network learns from the sole supervision of Z which is done once per sequence. X is the instance image. Output response map predicts the possible target location based on the highest response score [9].

A very detailed description of Siamese architecture along with mathematical discussion can be found in [9, 4].

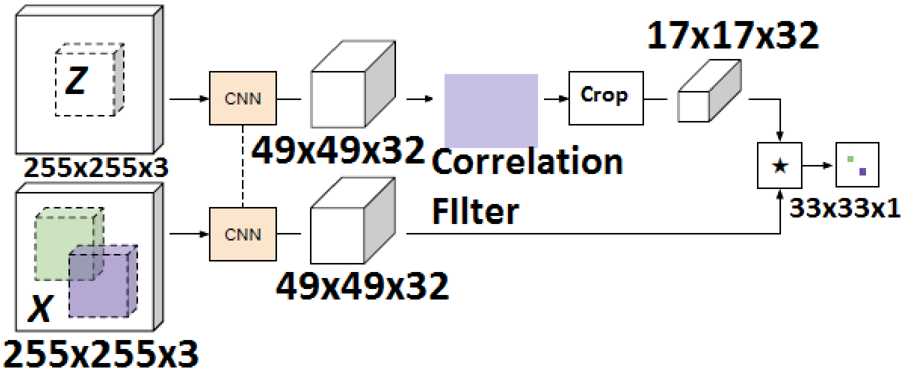

Figure 3.2: Correlation filter network. An additional correlation filter block in exemplar branch brings robustness to translation. End-to-end training model of a correlation filter which uses CNN features [18].

3.2.2 Correlation Filter Network

Correlation filter network(CFnet) is a modification over baseline Siamese network as given in equation (3.1). The CFnet architecture is shown in Figure 3.2. The new similarity measure function is given in equation (3.2).

$$h(z, x) = sW(\phi(z)) * \phi(x) + b \tag{3.2}$$

where s and b are scale and bias respectively. The correlation filter block $W(x)$ uses feature map $\phi(z)$ to learn a template by solving ridge regression problem in the Fourier domain [6]. The effect of circular boundaries has been mitigated by multiplying $\phi(z)$ with a cosine window and cropping the final template. Unlike SiameseFC, the appearance model in CFnet is updated after each frame using a rolling average in order to avoid abrupt transitions from frame to frame. The rest of the procedure is similar to that of the baseline Siamese as illustrated in Section 3.2.1. A deep insight into CFnet including back propagation can be found in [18].

3.2.3 Displacement Consistency

In order to avoid the target position deviating much from its previous position, most trackers [9, 18] employ a target centroid update strategy as shown in Figure 3.3. Mathematically, this update strategy can be written as in equation (3.3). This is usually employed to

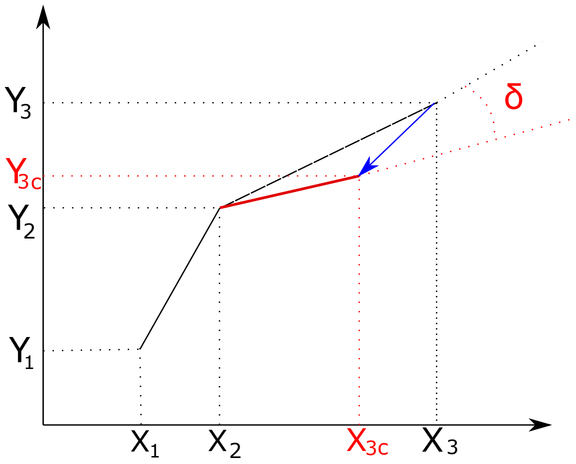

Figure 3.3: Conventional target centroid update strategy. Let $[X_1, Y_1]$ and $[X_2, Y_2]$ represent the target centroids in the first and second frame respectively. Let $[X_3, Y_3]$ represent the predicted centroid in the third frame. Let $[X_{3c}, Y_{3c}]$ represents the updated centroid in the third frame. Let δ represents the angular deviation occurred due to conventional centroid update.

enforce smoothness on the object motion. However, we identify that this is not sufficient to enforce the smoothness on the displacement (direction and distance) of the object in motion which can contribute to more reliable tracking. This can be noticed by observing the Figure 3.9. Due to the incremental angular deviation δ, the target centroid keeps drifting away from the actual centroid which decreases the overlap ratio. It is observed from Table 5.1 that minimizing this deviation δ as proposed has increased the success as well as precision.

$$[X_{3c}, Y_{3c}] = w \times [X_2, Y_2] + (1 - w) \times [X_3, Y_3] \tag{3.3}$$

We have integrated a new displacement consistency approach on top of the conventional approach to enhance the degree of smoothness. The angle consistency is illustrated in Figure 3.4. In distance consistency, the algorithm remembers the previous distance encountered and updates the new distance using rolling average. A pictorial representation of distance consistency is elucidated in Figure 3.5. After displacement consistency, the new centroid of the target is computed using equation (3.6). The accuracy and robustness after these integrations along with original values have been provided in Section 5 which proves the efficacy of the scheme.

$$\theta_{1n} = w_\theta \times \theta_0 + (1 - w_\theta) \times \theta_1 \tag{3.4}$$

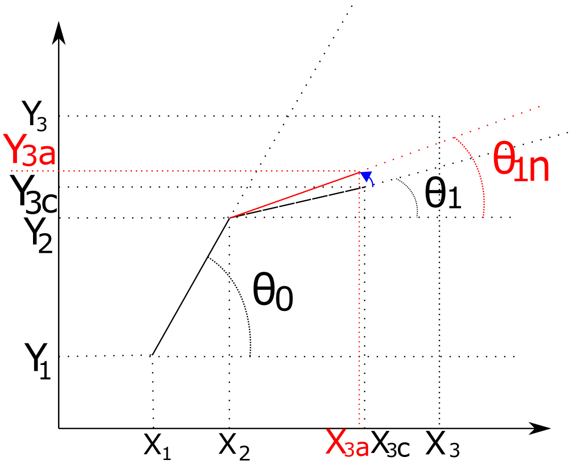

Figure 3.4: Angle consistency. Let θ_1 represents the angle of the centroid in the third frame with respect to the centroid in the second frame. Let θ_0 represents the angle of the centroid in the second frame with respect to the first. Let θ_{1n} represents the updated angle in the third frame. Let $[X_{3a}, Y_{3a}]$ represents the new updated centroid by using equation (3.4) with 1% weight given to previous angle i.e. w_θ = 0.01.

$$d_{1n} = w_d \times d_0 + (1 - w_d) \times d_1 \tag{3.5}$$

$$[X_{3n}, Y_{3n}] = [X_2, Y_2] + d_{1n}\angle\theta_{1n} \tag{3.6}$$

3.2.4 Scale Consistency

The conventional approach to estimate size of the target object is to form a scale pyramid and compute response map using each of these images [17]. The corresponding scale of the response map having maximum response score among all these response maps determines the size of the target object. Then that particular response map is used for obtaining the target centroid. In this standard approach, only the winning response map i.e the map having maximum response score among all maps decides the size of the object. However, as we know that in real scenarios, the scale of the object doesn't undergo drastic change from frame to frame as the scale change depends on the distance of the object from camera and as the objects move smoothly in many real scenarios. Though there are methods which add a penalty factor to the new target size, this applies only to the size of the target object. However, if the position of the target centroid itself is corrupted due to the use of wining response map only, it will persist in subsequent frames. In this standard scenario,

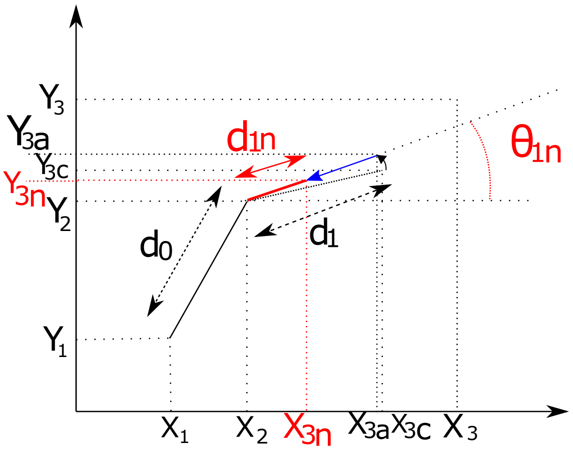

Figure 3.5: Distance consistency. Let d_0 represents the distance of centroid from frame 1 to 2. Let d_1 represents the distance from frame 2 to 3 after angle consistency. Let d_{1n} represents the updated distance obtained by using equation (3.5) with 1% preference given to previous distance i.e. w_d = 0.01. Let $[X_{3n}, Y_{3n}]$ represents the final position of the centroid after Displacement consistency.

the response maps that correspond to different scales aren't used in determining the centroid. Therefore, we propose to use Gaussian weighted average response map centred at the winning map and have variance as an additional hyper parameter. In this way we can incorporate the response maps that correspond to various scales in the scale pyramid. Our approach has enhanced the accuracy as well as robustness of the considered base trackers. The results of this experiment are described in details in Section 5. The pseudo code for Gaussian weighted average response map is provided in the following algorithm.

Algorithm: Scale Consistency using Gaussian weights

1. **Input parameters :** Let *responseMaps* represents the stack of response maps at each scale. μ represents the index of the winning response map. σ_{scale} represents the standard deviation of Gaussian weights. *scaleBins* numerically represents each scale i.e. *scaleBins(1)* represents the first scale, *scaleBins(2)* represents the second scale and so on. Let N represents the total number of scales used in the scale pyramid.

2. **Computation of scale weights and updation of *responseMap*:**

(a) Define weights for each scale as

$$scaleWeights = \frac{1}{\sqrt{2 \times \pi} \times \sigma_{scale}} \exp^{-(\frac{scaleBins - \mu}{\sigma_{scale}})^2}$$

(b) $responseMap =$

$$\sum_{i=1}^{N} [responseMaps(i) \times scaleWeights(i)]$$

3. **Output response map :** The output of this algorithm is the Gaussian weighted average *responseMap*.

3.2.5 Rotation Invariance

In this Section 3.2.5, we will discuss two different ways of incorporating rotation adaptiveness in tracking algorithms such as the proposed rotation invariant SiameseFC 3.2.5.1 and rotation invariant CFnet 3.2.5.2. The former can be used where the target object is not updated after each frame and the later can be used where the object is updated after each frame.

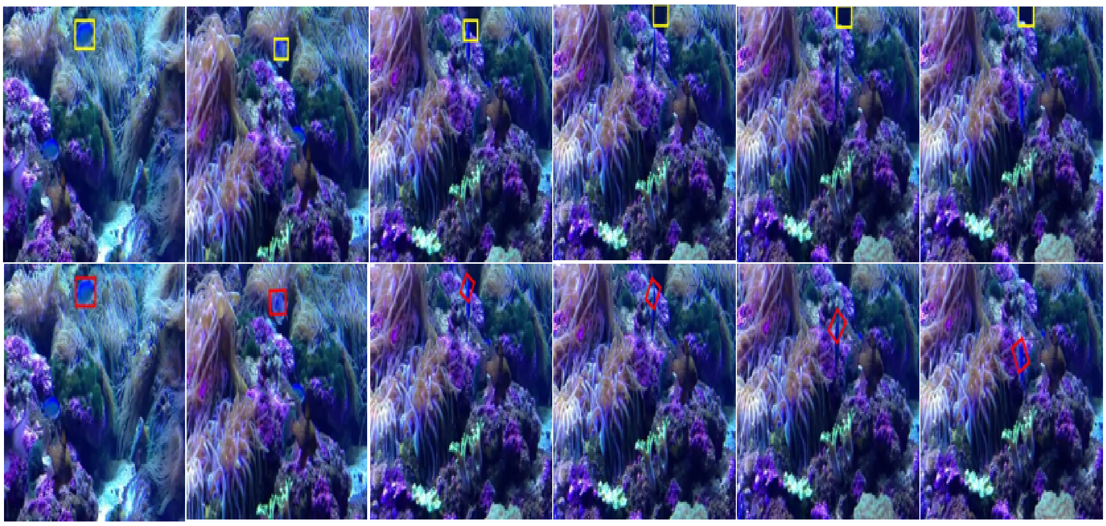

Figure 3.6: Sample frames from fish1 sequence denoted as one of the toughest sequences according to VOT 2016 results [5]. First column indicates the ground truth bounding box in the first frame. Our modified SiameseFC(red) successfully tracks the geometric deformations unlike original SiameseFC [9](yellow).

3.2.5.1 Rotation Invariant SiameseFC

When an object is in motion it can assume any of its rotated form or view from frame to frame. However, conventionally only a base template with a fixed (zero) orientation is employed to find the similarity. To incorporate robust RI tracking, we propose to augment various possible rotated images of the object and measure similarity with all these rotated images. The working of rotation invariant Siamese fully convolutional network has been illustrated in Figure 3.7. Since the appearance model isn't updated during tracking, the corresponding features of rotated exemplar can be extracted once for a sequence. Since the angle of rotation does not change drastically from frame to frame, only 5 nearest neighbour response maps are used in computing response map. The mean of the Gaussian weights is considered as the index of the winning response map and variance is tuned as an additional hyper-parameter in the similar manner as explained for different scales in Section 3.2.4. In order to avoid false alarm, we have computed three Gaussian weighted average response maps centred at top three maps according to their response scores. Accordingly there would be three most probable target centroids, out of which the final centroid is selected based on the highest score to displacement ratio in a sense that the object wouldn't have travelled far from its previous location. In this approach, as the path with dominant direction is detected, the bounding box can be rotated accordingly to increase the overlap ratio. A comparison between SiameseFC and rotation invariant SiameseFC is shown in Figure 3.6. We have evaluated our rotation invariant SiameseFC on VOT datasets [5] and the obtained results are provided in Section 5.

3.2.5.2 Rotation Invariant CFnet

Unlike rotation invariant SiameseFC, in rotation invariant CFnet the exemplar is updated after every frame [18]. In the second frame the object itself would have undergone some rotation. So there is no need to extract features from all the rotated exemplars beforehand, instead only forward and backward rotations after each model update would suffice. Therefore, there is no need to feed the angle of rotation back. Thus, there would be three response maps corresponding to each of the three rotations. Since we have only three response maps corresponding to three nearest neighbour angles and each response map

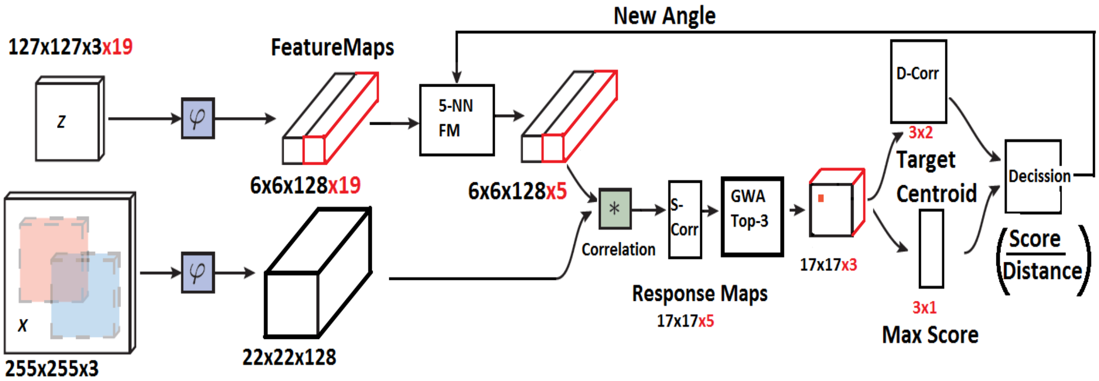

Figure 3.7: Rotation Invariant Siamese Fully Convolutional Network. The conventional SiameseFC extracts features for cropped exemplar, instead exemplar image can be rotated uniformly from -180° to 180° at an interval of θ and corresponding features can be extracted. Here, θ = 20°. So there would be 19 feature maps instead of only one. Assume initial newAngle to be 0°. 5-NN FM block passes 5 nearest neighbour feature maps based on the new angle of rotation. Let S-Corr and D-Corr blocks represent scale Section 3.2.4 and displacement Section 3.2.3 corrections respectively. GWA block computes Gaussian weighted average response map centred at top 3 maximum score response maps Section 3.2.4. Decision block computes the ratio of maximum score i.e. probability of detection and the corresponding distance from the previous location. The maximum ratio determines the final target centroid and the new angle of rotation which is determined as the angle corresponding to maximum ratio.

itself is a Gaussian weighted average map performed by the S-Corr block, there is no need to compute average of the three rotated maps again. The final centroid is computed by using the map having highest response score among all the three. In fact, using Gaussian weighted average after scale correction doesn't seem to improve the performance much, though it would be useful when more nearest neighbour rotated exemplars are used. This is a general approach which can be integrated into any state-of-the-art trackers to enhance their performance further. The rotation invariant CFnet is illustrated in Figure 3.8. The results of our proposed CFnet DS and CFnet DSR can be found in Section 5. In rest of the sections, we have referred CFnet-conv2 [18] as original CFnet and applied our consistency strategy to this model. We have used CFnet-conv2 in our experiments because it has less than 4% parameters used in five-layer baseline and outperforms the rest in the series [18].

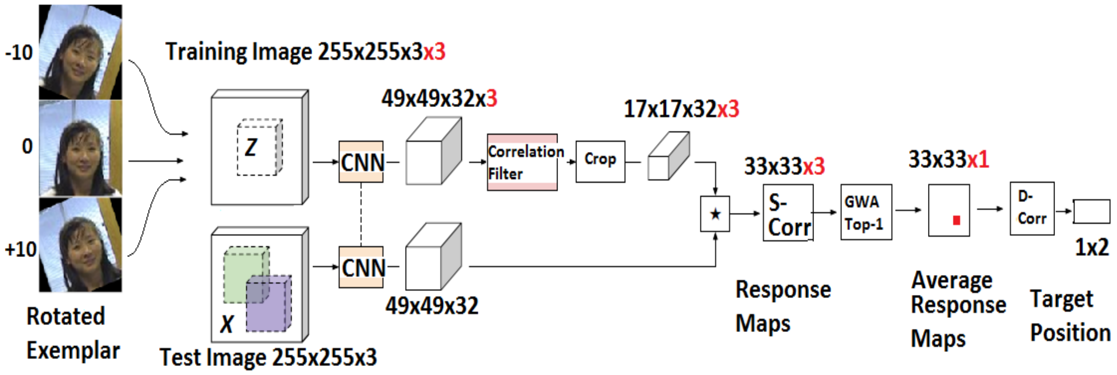

Figure 3.8: Rotation Invariant Correlation filter network. The input exemplar image is rotated by $\theta = [-\zeta, 0°, +\zeta]$. Here, $\zeta = 10°$ represents the angle of rotation of the exemplar. The angle of rotation 0° represents the actual cropped exemplar image obtained after each iteration. Thus, the three feature maps of rotated exemplar are correlated with the feature map of instance image which produce three most probable response maps. Let S-Corr and D-Corr blocks represent scale Section 3.2.4 and displacement Section 3.2.3 corrections respectively. The S-Corr block performs scale correction on these three response maps. The GWA block computes Gaussian weighted average response map centred at the winning response map. The D-Corr block performs displacement correction and computes the final target centroid.

3.3 Contributions in Correlation Filter based Trackers

3.3.1 Illumination Correction (IC) Filter

Illumination changes occur in a video due to dynamically changing environmental conditions, such as waving tree branches, low contrast regions, shadows of other objects, changes in object orientation relative to light sources etc. This variable illumination gives rise to low frequency interference, which is one of the prominent causes of disturbing the object's appearance. As the appearance of an object changes dramatically under different lighting conditions, the learned model fails to detect the object, leading to reduction in accuracy and robustness. Also, we may sometimes be interested in high frequency variations, such as edges, which are part of the dominant features in representing an object. Though these issues are investigated extensively in image processing community, to our knowledge, necessary attention for the same is not paid explicitly, even in the state-of-the-art trackers. Therefore, we intend to introduce Illumination Correction filter (IC) in the tracking paradigm in order to tackle the aforementioned issues up to some degree. At first, we employ a standard contrast stretching mechanism [75] to adjust the intensities of each

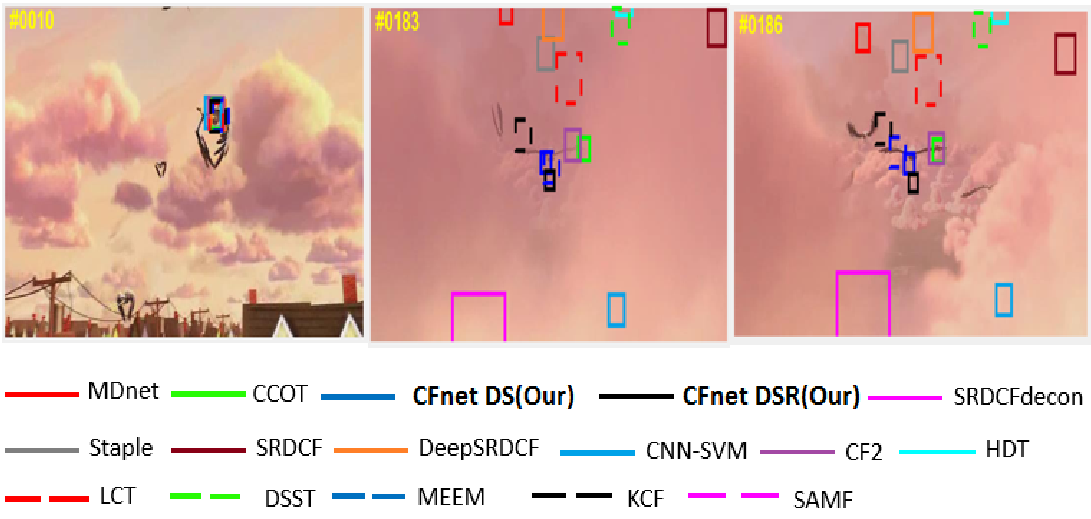

Figure 3.9: Sample frames from Bird1 sequence, one of the toughest sequences in OTB50 [14]. The results are obtained using fully integrated OTB toolkit. Our trackers have not deviated much from the target centroid mainly due to the integration of displacement correction.

frame. The contrast stretched image is then subjected to unsharp masking [75], a popular image enhancement technique in order to suppress the low frequency interference, and enhance high variations. To our surprise, the performance of the baseline trackers improves by a considerable amount just by enhancing the input images, as can be inferred from experimental results (Section 5). This validates the fact that the robust feature extractors still lack high quality visual inputs, which otherwise can lead to substantial gain in performance. To qualitatively assess the impact of IC, we have visualized the results of SRDCF and I-SRDCF in Figure 3.10.

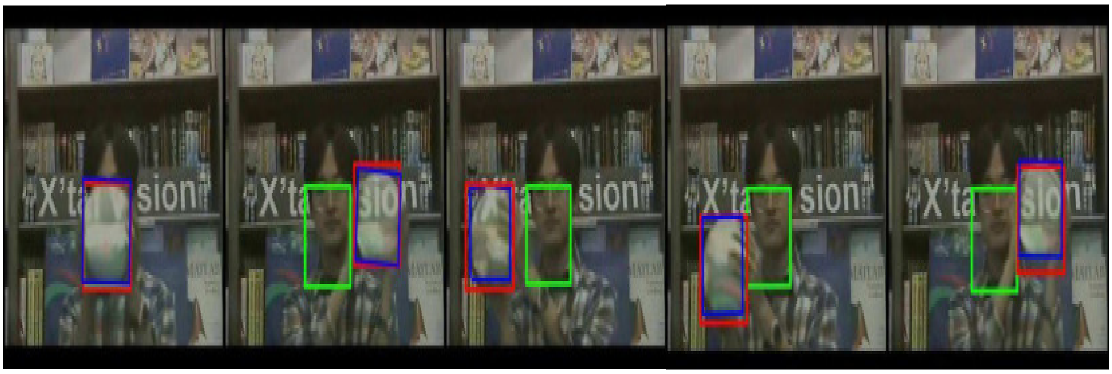

Figure 3.10: Sample frames from the sequence sphere of VOT2016 [5]. The blue, green, and red rectangle shows the groundtruth, SRDCF, and I-SRDCF (with IC) outputs, respectively.

3.3.2 Standard SRDCF Training and Detection

For the ease of disntinguishing and clearly understanding our contributions, we have used identical notations as in SRDCF [28]. In the standard DCF formulation, a multi-channel correlation filter f is learned from a set of training samples $\{(x_k, y_k)\}_{k=1}^{t}$. Each training sample x_k has a d-dimensional feature map, which is extracted from an image region. All the samples are assumed to be of identical spatial resolution $M \times N$. Thus, we have a d-dimensional feature vector $x_k(m,n) \in \mathbb{R}^d$ at each spatial location $(m,n) \in \Omega := \{0,\dots,M-1\} \times \{0,\dots,N-1\}$. We also denote feature layer $l \in \{1,\dots,d\}$ of x_k by x_k^l. The target of each training sample x_k is denoted as y_k, which is a scalar valued function over the domain Ω. The correlation filter f has a stack of d layers, each of which is a $M \times N$ convolution filter f^l. The response of the convolution filter f on a $M \times N$ sample x is computed by,

$$S_f(x) = \sum_{l=1}^{d} x^l * f^l. \tag{3.7}$$

Here, $*$ represents circular convolution. The desired filter f is obtained by minimizing the L^2-error between convolution response $S_f(x_k)$ of training sample x_k and the corresponding label y_k with a more general Tikhonov regularizer $w : \Omega \to \mathbb{R}$,

$$\varepsilon\left(f\right) = \sum_{k=1}^{t} \alpha_k \left\|S_f(x_k) - y_k\right\|^2 + \sum_{l=1}^{d} \left\|\frac{w}{MN} \cdot f^l\right\|^2. \tag{3.8}$$

Here, $\cdot$ denotes point-wise multiplication. With the help of Parseval's theorem, the filter f can be equivalently computed by minimizing the equation (3.8) in the Fourier domain with respect to Discrete Forurier Transform (DFT) coefficients $\hat{f}$,

$$\hat{\varepsilon}(\hat{f}) = \sum_{k=1}^{t} \alpha_k \left\|\sum_{l=1}^{d} \hat{x}_k^l \cdot \hat{f}^l - \hat{y}_k\right\|^2 + \sum_{l=1}^{d} \left\|\frac{\hat{w}}{MN} * \hat{f}^l\right\|^2. \tag{3.9}$$

Here, $\hat{}$ denotes the DFT of a function. After learning the DFT coefficients $\hat{f}$ of filter f, it is typically applied in a sliding-window-like manner on all cyclic shifts of a test sample z. Let $\hat{s} := \mathcal{F}\{S_f(z)\} = \sum_{l=1}^{d} \hat{z}^l \cdot \hat{f}^l$ denote DFT ($\mathcal{F}$) of the convolution response $S_f(z)$ evaluated at test sample z. The convolution response $s(u,v)$ at continuous location

$(u, v) \in [0, M) \times [0, N)$ are interpolated by,

$$s(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \hat{s}\,(m,n)\, e^{i2\pi\left(\frac{m}{M}u + \frac{n}{N}v\right)}. \tag{3.10}$$

Here, i denotes the imaginary unit. The maximal sub-grid location (u^{*}, v^{*}) is then computed by optimizing arg $\max_{(u,v)\in[0,M)\times[0,N)} s\,(u,v)$ using Newton's method, starting at maximal grid-level score $(u^{(0)}, v^{(0)}) \in \Omega$. Simillar to [17], the SRDCF tracker applies sub-grid interpolation at multiple resolutions z_r and computes corresponding detection scores s_r in order to estimate scale efficiently. In a nutshell, the standard SRDCF adapts translation invariance efficiently by exploiting the periodic assumption with spatial regularization, but this does not learn rotation adaptiveness inherently. Therefore, we propose to extend the discriminative power of SRDCF by learning rotation adaptive filters.

3.3.3 Rotation Adaptive Correlation Filters in Tracking

We propose to incorporate rotation adaptiveness in spatially regularized correlation filters by learning from appropriately oriented training samples. Similar to SRDCF, we solve the resulting optimization problem in the Fourier domain, by employing a deterministic orientation in each training sample. Let θ_k denotes the orientation corresponding to x_k. Without loss of generality, it can be assumed that $\theta_k = 0, \forall k \leq 1$. The training sample x_k undergoes rotation θ_k by,

$$x_k^{\theta}\,(m,n)_{(m,n)\in\Omega} = \begin{cases} x_k\,(m',n') & , \quad (m',n') \in \Omega \\ 0 & , \quad elsewhere \end{cases} \tag{3.11}$$

where (m, n) and (m', n') are related by,

$$\begin{bmatrix} n \\ m \end{bmatrix} = \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{bmatrix} \begin{bmatrix} n' \\ m' \end{bmatrix}. \tag{3.12}$$

In other words, x_k^{θ} is obtained by rotating x_k anti-clockwise with an angle θ_k in the euclidean space and cropping same size $M \times N$ as x_k. In order to avoid wrong gradient estimation due to zero paddings, we use a common solution that bands the rotated image

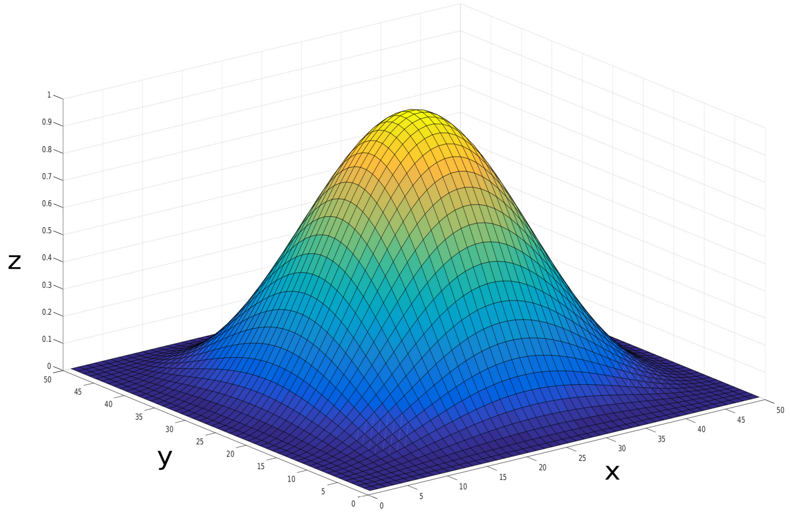

Figure 3.11: Visualization of the cosine window used to suppress the false gradient estimation due to rotated training samples.

patch with cosine window. For the ease of understanding, we have visualized the cosine window used in Figure 3.11. This does not disturb the object structure assuming that the patch size is larger than the target object. This is different from standard SRDCF, in a sense that we learn the multi-channel correlation filter f from properly oriented training samples $\left\{\left(x_k^{\theta}, y_k\right)\right\}_{k=1}^{t}$. The training stage of rotation adaptive filters is explained in the following Section 3.3.3.1.

3.3.3.1 Training

The convolution response $S_f(x_k^{\theta})$ of the rotated training samples $x_k^{\theta} \in \mathbb{R}^d$ are computed by,

$$S_f(x_k^{\theta}) = \sum_{l=1}^{d} x_k^{\theta l} * f^l. \tag{3.13}$$

After incorporating rotation into the DCF formulation, the resulting cost function is expressed as,

$$\varepsilon_{\theta}\left(f\right) = \sum_{k=1}^{t} \alpha_k \left\| S_f(x_k^{\theta}) - y_k \right\|^2 + \sum_{l=1}^{d} \left\| \frac{w}{MN} \cdot f^l \right\|^2. \tag{3.14}$$

Similar to SRDCF, we perform the Gauss-Seidel iterative optimization in Fourier domain by computing DFT of equation (3.14) as,

$$\hat{\varepsilon}_\theta(\hat{f}) = \sum_{k=1}^{t}\alpha_k \left\| \sum_{l=1}^{d} \hat{x}_k^{\theta l}\cdot \hat{f}^l - \hat{y}_k \right\|^2 + \sum_{l=1}^{d}\left\| \frac{\hat{w}}{MN} * \hat{f}^l \right\|^2 . \tag{3.15}$$

The equation (3.15) is vectorized and simplified further by using fully vectorized real-valued filter, as implemented in the standard SRDCF [28]. The aforementioned training procedure is feasible, provided we obtain the object orientation corresponding to all the training samples beforehand. In the following Section 3.3.3.2, we propose an approach to detect the object orientation by optimizing an additional objective function.

3.3.3.2 Detection

At the detection stage, the correlation filter f learned from t training samples are utilized to compute the convolution response of a test sample z obtained from $(t+1)^{th}$ frame, which is then optimized to locate the object in that $(t+1)^{th}$ frame. For example, at $t = 1$, we learn the coefficients of f from $(x_{k=1}^{\theta=0^\circ}, y_{k=1})$ and detect the object location, (u_{k+1}^*, v_{k+1}^*), and orientation, θ_{k+1} in the $(t+1)^{th}$, i.e., 2^{nd} frame. For efficient detection of scale, we construct different resolution test samples $\{z_r\}_{r\in\left\{\left\lfloor\frac{1-S}{2}\right\rfloor,\dots,\left\lfloor\frac{S-1}{2}\right\rfloor\right\}}$ by resizing the image at various scales a^r, as implemented in SRDCF[28]. Here, S and a denote the number of scales and scale increment factor, respectively. Next, we discuss the false positive elimination scheme, which offers notable gain in the overall performance.

3.3.3.3 False Positive Elimination (FPE)

As per our extensive experiments, we report that the convolution response map of test sample may sometimes contain multiple peaks with equal detection scores. This situation usually arises when the test sample is constructed from an image region that consists of multiple objects with similar representations as target object. In fact, this issue can occur in many real world scenarios, such as glove, leaves, rabbit etc. sequences from VOT2016 dataset [5]. Therefore, we propose to maximize $\frac{s(u,v)}{\left\|\left(u-u_k^*,v-v_k^*\right)\right\|}$ unlike SRDCF, which focuses on maximizing $s(u,v)$ alone. Here, (u_k^*, v_k^*) denote the sub-grid level target location in the k^{th} frame. Thereby, we intend to detect the object that has high response score as well as minimum deviation from previous location. Arguably, this hypothesis is justified by the fact that it is less likely for an object to undergo drastic deviation from immediate past location. Thus, the issue of multiple peaks in convolution response is eliminated up to some extent, as shown in Figure 3.12.

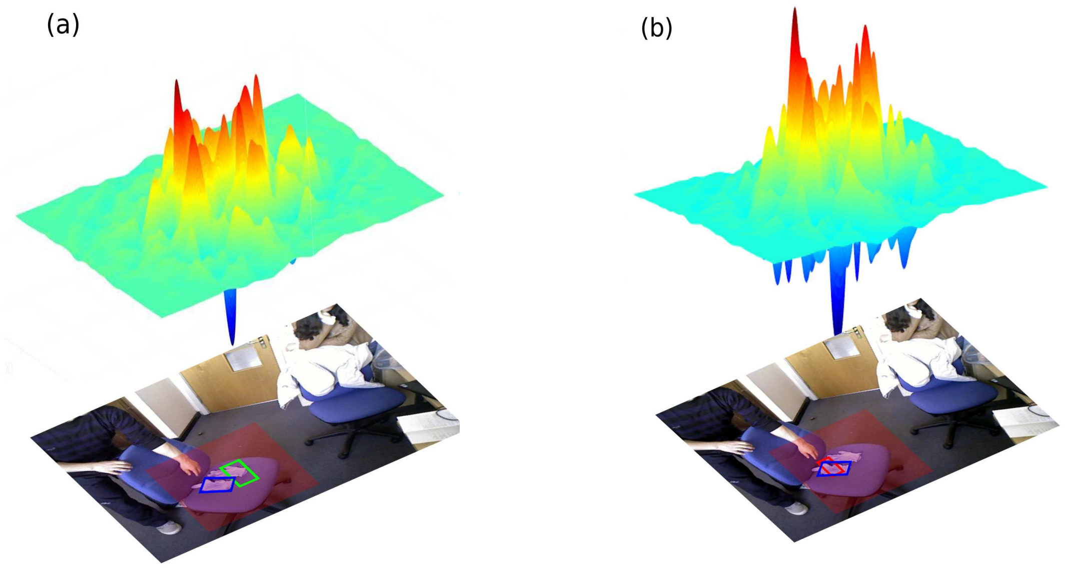

Figure 3.12: Sample frames from the sequence glove of VOT2016 [5]. The blue, green, and red rectangle shows the groundtruth, ECO, and F-ECO outputs, respectively. Convolution response of shaded (red) region (a) without, and (b) with false positive elimination.

3.3.3.4 Detection of Orientation (DoO)

Here, we elaborate the detection mechanism of object's orientation in the test sample. Let $\hat{s}_\theta := \mathcal{F}\left\{S_f(z^\theta)\right\} = \sum_{l=1}^{d} \hat{z}^{\theta l} \cdot \hat{f}^l$ represents the DFT of convolution response $S_f(z^\theta)$, evaluated at θ orientation of test sample z. Similar to equation (3.10), we compute $s_\theta(u,v)$ on a coarse grid $(u,v) \in \Omega$ by,

$$s_\theta(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \hat{s}_\theta\left(m,n\right) e^{i2\pi\left(\frac{m}{M}u + \frac{n}{N}v\right)}. \tag{3.16}$$

Then, the aim is to find orientation that maximizes the total energy content in the convolution response map by,

$$\theta_{k+1} = \text{arg max}_{\theta \in \Phi} \left\{ \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \left(\frac{S_{\theta}\left(u, v\right)}{\|(u - u_k^*, v - v_k^*)\|} \right)^2 \right\}. \tag{3.17}$$

Here, $\Phi := \{\theta_k \pm a\delta\}$, where $a = 0, 1, 2, \ldots, A$. Thus, the orientation space Φ consists of $(2A + 1)$ number of rotations with step size δ. In our experiments, we have used $\delta = 5^{\circ}$, and $A = 2$ based on the fact that an object's orientation is less likely to change drastically between consecutive frames. Nevertheless, the orientation can be further optimized by Newton's approach, or any suitable optimization algorithm, starting at optimal coarse orientation θ_{k+1}. Also, a suitable combination of A and δ can be chosen for searching exhaustively in Φ, but at the expense of time complexity. Next, we incorporate the FPE and DoO techniques in Fast Sub-grid Detection method of standard SRDCF (Section 3.3.3.5) formulation.

3.3.3.5 Fast Sub-grid Detection

We apply the Newton's optimization strategy, as in SRDCF, for finding the sub-grid location that maximizes the detection score. However, we incorporate the false positive elimination and optimal orientation in the standard SRDCF sub-grid detection. Thus, we compute the sub-grid location that corresponds to maximum detection score by,

$$\left(u_{k+1}^*, v_{k+1}^*\right) = \text{arg max}_{(u,v) \in [0,M) \times [0,N)} \left\{ \frac{S_{\theta_{k+1}}(u, v)}{\|(u - u_k^*, v - v_k^*)\|} \right\}, \tag{3.18}$$

starting at $(u^{(0)}, v^{(0)}) \in \Omega$, such that $\left\{ \frac{S_{\theta_{k+1}}(u^{(0)}, v^{(0)})}{\left\|\left(u^{(0)} - u_k^*, v^{(0)} - v_k^*\right)\right\|} \right\}$ is maximal.

3.3.4 Displacement Consistency

Motivated by the displacement consistency techniques, as proposed in [35], we enhance the degree of smoothness imposed on the movement variables, such as speed and angular displacement. We update the sub-grid location, $\left(u_{k+1}^*, v_{k+1}^*\right)$ obtained from equation (3.18) by,

$$\begin{aligned}\left(u_{k+1}^{*}, v_{k+1}^{*}\right) &= \left(u_{k}^{*}, v_{k}^{*}\right) + d_{1n}\angle\varphi_{1n}, \\ d_{1n} &= \omega_d \times d_1 + (1-\omega_d) \times d_0, \\ \varphi_{1n} &= \omega_a \times \varphi_1 + (1-\omega_a) \times \varphi_0,\end{aligned} \tag{3.19}$$

where, $d_0 = \left\|\left(u_k^* - u_{k-1}^*, v_k^* - v_{k-1}^*\right)\right\|$, $d_1 = \left\|\left(u_{k+1}^* - u_k^*, v_{k+1}^* - v_k^*\right)\right\|$, $\varphi_0 = \arctan\left(u_k^* - u_{k-1}^*, v_k^* - v_{k-1}^*\right)$, $\varphi_1 = \arctan\left(u_{k+1}^* - u_k^*, v_{k+1}^* - v_k^*\right), \omega_d = 0.9, \omega_a = 0.9$. The abrupt transition from (u_k^*, v_k^*) to $\left(u_{k+1}^*, v_{k+1}^*\right)$ is restricted up to some extent by reducing the contribution of d_1 and φ_1 slightly to 0.9. For $\omega_d = \varphi = 1$, the updated $\left(u_{k+1}^*, v_{k+1}^*\right)$ of equation (3.19) remains unaltered from the optimal solution of equation (3.18).

3.4 Contributions in Regression based Trackers

The overall architecture of our Temporal Regression (TR) based integration is shown in Figure 3.13. As discussed in the contributions and the preceding sections we enhance the visual inputs before feature extraction through an EF (Section 3.4.2), and thereafter, the essential processing required for TR (Section 3.4.3) is depicted. For the sake of experimentation, we integrate the proposed methodology in ECO tracker, and showcase the efficacy by comparing with various state-of-the-art trackers on OTB50 and VOT2016 datasets. We specifically provide a systematic approach based on well known regularization framework for incorporating temporal information in DCF based trackers. The framework has the provision to provide the proportionate weight-age across the previous frames based on their similarity with current frame and also consider feature prioritization based on the average information content in temporal domain.

At the beginning, we apply EF to each frame. After enhancement of visual information (each frame), the search region from each frame is fed to the feature extractor. The search region is decided based on the previous position and scale as implemented in [38]. The high dimensional CNN features, as extracted in [38], are projected onto a low dimensional space, aiming at reduction of time complexity. To achieve this, we have applied

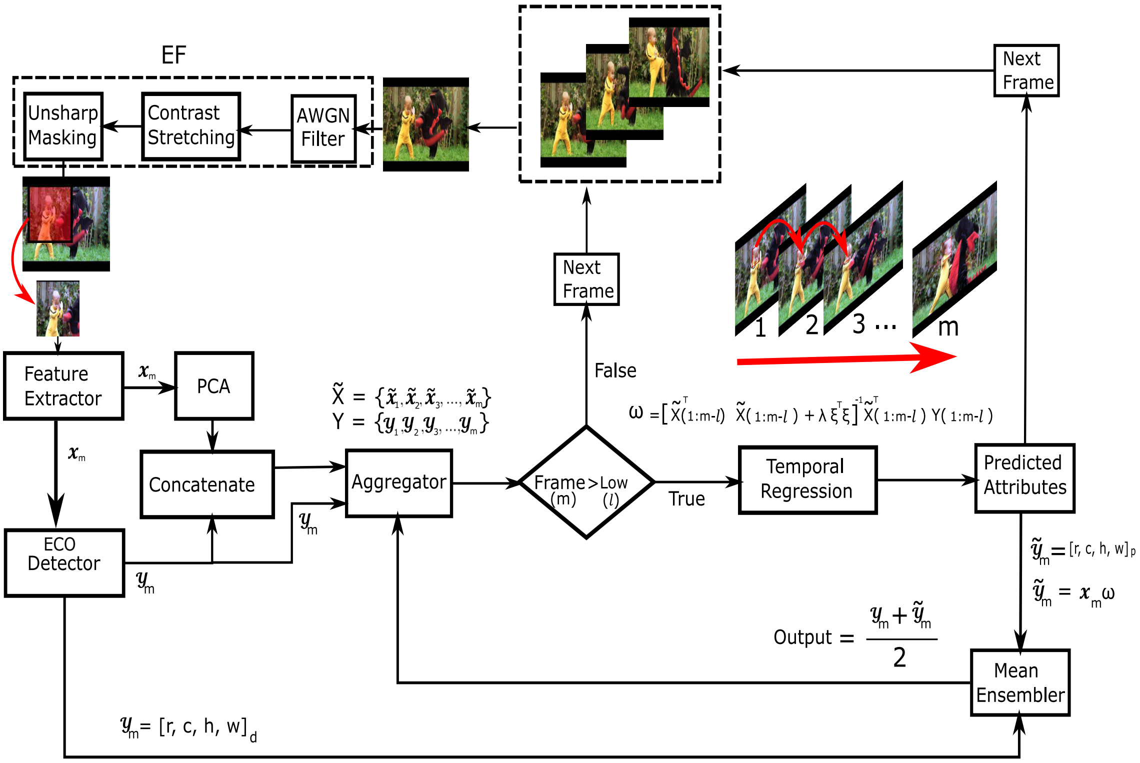
EF
Unsharp Masking
Contrast Stretching
AWGN Filter
Next Frame
Next Frame
1
2
3 ...
m
Feature Extractor
x_m
PCA
x_m
$\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, ..., \tilde{x}_m\}$
$Y = \{y_1, y_2, y_3, ..., y_m\}$
False
$\omega = [\tilde{X}^T(1:m-l)\ \tilde{X}(1:m-l) + \lambda \xi^T \xi]^{-1} \tilde{X}^T(1:m-l)\ Y(1:m-l)$
Concatenate
Aggregator
Frame (m) > Low (l)
True
Temporal Regression
Predicted Attributes
ECO Detector
y_m
y_m
$\tilde{y}_m = [r, c, h, w]_p$
$\tilde{y}_m = x_m \omega$
Output = $\frac{y_m + \tilde{y}_m}{2}$
Mean Ensembler
$y_m = [r, c, h, w]_d$

Figure 3.13: Temporal regression with weighted aggregation and enhancement filter as proposed in this thesis (Section 3.4). Each frame is passed through enhancement filter (EF) before going into feature extractor. The (ECO) detector uses the extracted features and predicts the target attributes based on spatial correlation. The extracted features are projected into a low dimensional space where these are concatenated with target attributes. The concatenated features are then aggregated based on temporal correspondence and used in learning the parameters (ω) of temporal regression.

principal component analysis (PCA) with 90% captured variance. The compressed features are then concatenated with ECO detector outputs, and thereafter, these concatenated features with weighted aggregation (Section 3.4.3.1) are accumulated in the aggregator.

Let X be a collection of feature vectors in m frames $\{x_1, x_2, ..., x_m\} \in \mathbb{R}^{1\times n}$, where n represents the number of features extracted from the highly correlated patch in each frame. Let Y be a collection of regression targets of the corresponding m frames $\{y_1, y_2, ..., y_m\} \in \mathbb{R}^{1\times p}$, where p represents the dimension of attributes in the order of target centroid $(row, column)$ and size $(height, width)$,i.e., (r, c, h, w). The matrix Y contains the output y_m of the detector and X contains the corresponding input features to the detector. For robust prediction of $\widetilde{y}_m = x_m\omega$, we learn the regressor parameters $\omega \in \mathbb{R}^{n\times p}$ by accumulating the previous estimates of target attributes $Y(1 : m-1)$, and the associated features with controlled suppression of uncorrelated frames $\widetilde{X}(1 : m-1)$. Then we propose to augment temporal regression output $\widetilde{y_m}$, with the spatial ECO detector output y_m, by considering mean ensemble $\frac{y_m+\widetilde{y_m}}{2}$ consistently. The ensemble attributes are then fed back to the aggregator, which are used to update the accumulated attributes in Y and X. However, updating target attributes in both Y and X may unfairly emphasize falsely tracked targets due to marginal inclusion of detector outputs. Therefore, we either update the concatenated detector outputs in X by $x_m(end-p-1:end) \leftarrow \frac{y_m+\widetilde{y_m}}{2}$ or regression targets in Y by $y_m \leftarrow \frac{y_m+\widetilde{y_m}}{2}$. This is indeed the case as our experiments show that updating X turns out to be more effective than the other counter parts. Primarily, we discuss briefly the architecture and fundamental working principles of ECO (Section 3.4.1), and thereafter, the detailed contributions as illustrated in Figure 3.13.

3.4.1 Baseline Approach: ECO

In this section, we briefly discuss the recently introduced Efficient Convolution Operators for Tracking (ECO) [38], which we have adopted as our baseline. The ECO tracker has demonstrated exceptional outcomes in various benchmark datasets including OTB and VOT. The introduction of factorized convolution operators in ECO, has reduced the parameters in the DCF model drastically. Apart from efficient convolution operators, the ECO tracker proposes a method for feasible memory consumption by reducing the number of training samples, while maintaining diversity. Moreover, the efficient model update strategy, as proposed in ECO, reduces the unfavourable sudden appearance changes as a result of illumination variation, out-of-view, and deformation. As per the comprehensive experimentation, the ECO tracker with deep features outperforms all the previous trackers that rely on DCF formulation. Motivated by these findings, we have integrated the proposed framework in baseline ECO with deep settings in light of further improvement, and demonstrated that the newly developed approach offers significant gain in numerous challenging sequences.

3.4.2 Enhancement Filter (EF)

In real world scenarios, it is intractable to obtain high quality visual information due to stochastic nature of the environment. To combat the assimilated assessment of several random fluctuations, while preserving the fine/sharp details of the information content in images, we employ edge adaptive Gaussian smoothing. The AWGN Filter block in Figure 3.13 represents edge preserved Gaussian smoothing of additive white Gaussian noise (AWGN) with three channel or 3D multi variate Gaussian kernel of standard deviation close to 0 each (here, 0.1), in order not to smooth the edges. A detailed description on AWGN filters can be found in [75]. To span the whole intensity from 0 to 255, while rectifying the contrast imbalance in each channel, we have employed linear contrast stretching to each frame after AWGN removal.

Low frequency interference arises when the visual information is gathered under variable illumination. This holds in almost all indoor scenes because of the inverse square law of light propagation. Arguably, the outdoor scenes do not suffer from this effect, because the sun is so far away, that all the tiny regions in an image appear to be at equal distance from it. However, other illuminating sources may produce low frequency interference in an unconstrained environment. Also, we may sometimes be interested in minute details of a scene, or scenes that manifest in high frequencies such as object boundaries. Therefore, it is often desirable to suppress the unwanted low frequencies to leverage high variations in a scene. While this issue has been studied extensively in image processing tasks [75], even in state-of-the-art trackers, as per our knowledge, the necessary attention for the same is not paid explicitly. So, we intend to introduce the popular algorithm, local unsharp masking on visual object tracking paradigm. In local unsharp masking, a local window is considered while computing a low pass filtered image. The low pass filtered image is then subtracted from the actual image and the difference is multiplied by an amplification factor. The difference is amplified only when it exceeds certain threshold ($= 0$, here), which is used to suppress high frequency fluctuations due to noise. Thus, the amplification factor is chosen according to the local variance. The aforementioned transformation can be achieved using the Eqn. (3.20). A detail description of these methods along with essential comparisons can be found in [75].

$$g(x,y) = A[f(x,y) - m(x,y)] + m(x,y) \tag{3.20}$$

where $A = \frac{kM}{\sigma(x,y)}$, k is a scalar, M is the average intensity of the whole image, $\sigma(x,y)$ represents variance of the window. $g(x,y)$, $f(x,y)$, and $m(x,y)$ represent resulting image, input image, and low pass version of $f(x,y)$, respectively.

3.4.3 Temporal Regression by Tikhonov Regularization in Tracking

Here, we elaborate our Temporal Regression (TR) framework with detailed analysis of each key components such as Weighted Aggregation, Feature Prioritization, Tikhonov Regularization, and Mean Ensembler.

3.4.3.1 Weighted Aggregation (WA) in Temporal Regression

Here, we illustrate the weighted aggregation strategy, which brings substantial gain on a diverse set of tough sequences from tracking benchmarks. Let $\alpha \in \mathbb{R}^{m\times 1}$ represent the coefficients for modulating the m frames in temporal domain. The elements of α are computed based on the projection of x_m onto X which consists of m vectors in n dimensional vector space. An important point to remember here is, even if m frames are modulated based on this correlation metric, the frame x_m remains unaltered due to maximal correlation, and also, it is excluded from training set (Section 3.4.3). The underlying hypothesis is to learn from the weighted aggregation of preceding features based on similarity measure with the test frame x_m, and predict the current attributes $\widetilde{y_m}$. Thereby, we inhibit the dominance of dissimilar frames in voting for target attributes in the current frame. In other words, features from only those frames are amplified which have a contextual correspondence with the test frame in the temporal domain. We squash the elements of α using sigmoid activation in order to map the correlation values to a fixed smooth range between 0 and 1 for all frames, reason for which is understandable. Thus, the modulation coefficients can be computed using Eqn. (3.21), as used in our simulations, or by Eqn. (3.22) to prevent early saturation. Early saturation can be detected by observing the correlation values. If the correlation values are very high, the sigmoid activation would give saturated coefficients for all, which will restrain the suppression of uncorrelated frames. In such cases, Eqn. (3.22) would be useful in discrimination.

$$\alpha = sigmoid(\frac{Xx_m^T}{n}), \tag{3.21}$$

$$\alpha = sigmoid(\sqrt{\frac{Xx_m^T}{n}}), \tag{3.22}$$

where $X \in \mathbb{R}^{m\times n}$, $x_m \in \mathbb{R}^{1\times n}$, and $\alpha \in \mathbb{R}^{m\times 1}$.

The features from preceding m frames are modulated by α to enhance the contribution of highly correlated frames, while suppressing the contribution of uncorrelated ones. Thereby, efficient aggregation of past information is utilized in learning the parameters of regressor, which leads to robust prediction of target attributes in the subsequent frames. The modulated training samples are computed by Eqn. (3.23).

$$\widetilde{X} = X. * \alpha \tag{3.23}$$

where $.*$ represents row wise multiplication with corresponding scalar value of α ,i.e., $\widetilde{X}(i,:) = X(i,:) * \alpha(i), i = 1, 2, \ldots, m$ and $*$ represents element wise multiplication. After obtaining $\widetilde{X} = \{\widetilde{x_1}, \widetilde{x_2}, \ldots, \widetilde{x_m}\}$, the training features are further regulated based on entropy of the associated random variables (Section 3.4.3.2).

3.4.3.2 Feature Prioritization through Entropy Estimation (FPEE)

In this section, we briefly discuss an efficient feature engineering approach as part of WA, taking into account the uncertainty preserved in each feature in the temporal domain. The hypothesis is to estimate the entropy of each feature in $\widetilde{X}$ across all m frames, and use this information content to enhance the contribution of that particular set of features towards estimation of target attributes. This can be achieved by modulating each column of $\widetilde{X}$, which is in contrast to row wise modulation, as done by α. Let $f_i \in \mathbb{R}^{1\times m}, i = 1, 2, \ldots, n$ represent a random variable with observations drawn from the i^{th} feature of all m frames. For the ease of experimentation, the observations of these random variables are used to

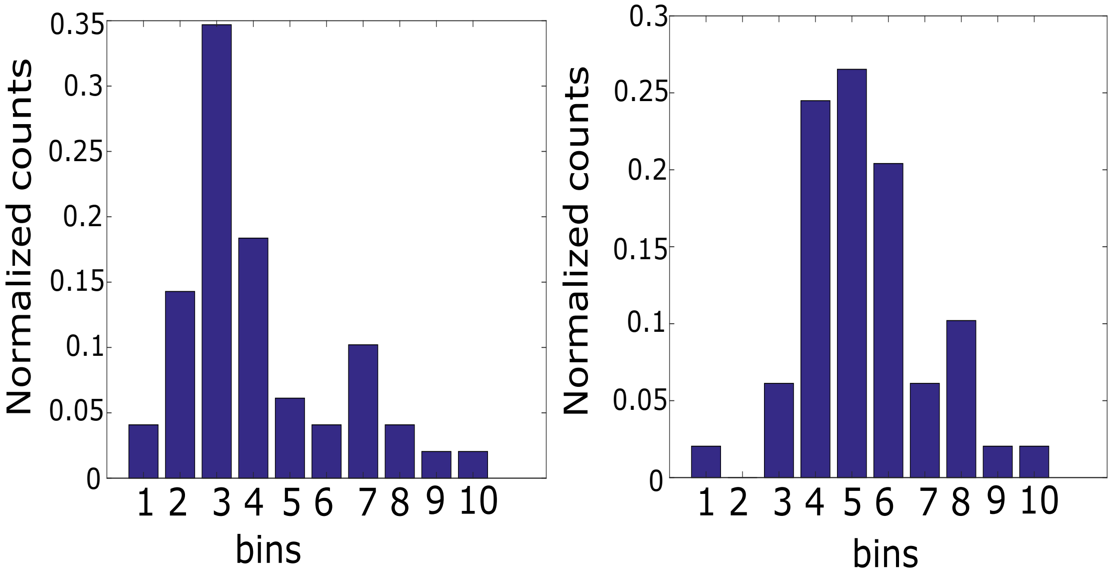

Figure 3.14: The histogram of features are computed with fixed number of bins(here, 10). The normalized count is used as probability density $\mathbb{P}_{f_i}$. The distributions of f_1(left) and f_{104}(right) are used to quantify the amount of information content in the corresponding features.

estimate the distribution based on normalized histogram counts. For better understanding, we have visualized the histogram of two random variables f_1 and f_{104} in Figure 3.14.

The basic intuition is, learning that an unlikely event has occurred is more informative than a likely event has occurred. Therefore, we define self-information of event $f = \mathrm{f}$ by $I(\mathrm{f}) = -\log \mathbb{P}_f(\mathrm{f})$, with base e, as characterized in information theory. The self-information deals with a single outcome which leads to several drawbacks, such as an event with unity density has zero self-information, despite it is not guaranteed to occur. Therefore, we have opted Shannon entropy,

$$H(f) = \mathbb{E}_{f \sim \mathbb{P}_f} \left[I(\mathrm{f}) \right] = -\mathbb{E}_{f \sim \mathbb{P}_f} \left[\log \mathbb{P}_f(\mathrm{f}) \right],$$

which is used to deal with such issues [76], to quantify the amount of uncertainty conserved in the entire distribution. We use this uncertainty measure to enhance, or suppress the training features in $\widetilde{X} = f_1, f_2, \ldots, f_n$ by Eqn. (3.24).

$$\widetilde{f_i} = f_i * H(f_i), i = 1, 2, \ldots, n \tag{3.24}$$

Consequently, the parameters (ω) of temporal regression are computed with the updated training features $\widetilde{X} = \left\{ \widetilde{f_1}, \widetilde{f_2}, \ldots, \widetilde{f_n} \right\}$.

3.4.3.3 Tikhonov Regularization in Temporal Regression

Here, we describe the context in which we employ standard Tikhonov regularization. To ensure smooth variation of temporal weights (ω), we have penalized the coefficients with larger norms. In our formulation, $\lambda\xi$ represents the standard Tikhonov operator. For equal preference, we have set ξ to be an identity matrix $I \in \mathbb{R}^{m\times n}$, and λ to be 1000. Thus, after incorporating temporal correspondence by WA and FPEE, the standard ridge regression has been updated to Eqn. (3.25).

$$J = \left\| \widetilde{X}\omega - Y \right\|_2^2 + \lambda \left\| \xi\omega \right\|_2^2 \tag{3.25}$$

The closed-form solution of J can be obtained as following.

$$\begin{aligned}
&\nabla_\omega \left\{ \left\| \widetilde{X}\omega - Y \right\|_2^2 + \lambda \left\| \xi\omega \right\|_2^2 \right\} = 0 \\
&i.e. \nabla_\omega \left\{ (\widetilde{X}\omega - Y)^T (\widetilde{X}\omega - Y) \right\} + \lambda \nabla_\omega \left\{ (\xi\omega)^T (\xi\omega) \right\} = 0 \\
&i.e. \nabla_\omega \left\{ \omega^T \widetilde{X}^T \widetilde{X}\omega - \omega^T \widetilde{X} Y - Y^T \widetilde{X}\omega + Y^T Y \right\} + \lambda \nabla_\omega \left\{ \omega^T \xi^T \xi\omega \right\} = 0 \\
&i.e. 2\widetilde{X}^T \widetilde{X}\omega - \widetilde{X}^T Y - \widetilde{X}^T Y + 2\lambda\xi^T\xi\omega = 0 \\
&i.e. \left[\widetilde{X}^T \widetilde{X} + \lambda\xi^T\xi \right] \omega = \widetilde{X}^T Y
\end{aligned} \tag{3.26}$$

$$\boxed{\omega = \left[\widetilde{X}^T \widetilde{X} + \lambda\xi^T\xi \right]^{-1} \widetilde{X}^T Y,}$$

where $\omega \in \mathbb{R}^{n\times p}$, and the predicted target attributes are computed by $\widetilde{y_m} = x_m\omega$.

3.4.3.4 Mean Ensembler for Spatio-Temporal Aggregation

This section depicts the theoretical background on the efficacy of mean ensemble. The proposed dynamic model comprises two models having minimal interdependence in their way of implementation. The detector works in the spatial domain with efficient training and robust model update strategy. On the contrary, the regression model operates in the temporal domain maximizing the correspondence with visual features from the current frame, and capturing the physically meaningful movement variables, such as position and angular displacement. Hence, the composition of these two models with bootstrap aggregation would be beneficial in lessening the overall error [76]. Assume there are k models with error $\delta_i \sim \mathcal{N}(\mu = 0, \sigma^2 = v), i = 1, 2, \ldots, k$. Let the covariance $\mathbb{E}[\delta_i \delta_j] = c$. The error made by the mean ensembler output would be $\frac{1}{k}\sum_{i=1}^{k} \delta_i$. The expected squared error predicted by the ensembler would be

$$\mathbb{E}\left[\left(\frac{1}{k}\sum_{i=1}^{k}\delta_i\right)^2\right] = \mathbb{E}\left[\frac{1}{k^2}\sum_{i=1}^{k}\left(\delta_i^2 + \sum_{j=1, j\neq i}^{k}\delta_i\delta_j\right)\right] = \frac{v}{k} + \frac{k-1}{k}c.$$

If the models are perfectly correlated ,i.e., $\mathbb{E}\left[\delta_i \delta_j\right] = c = v$, then there will not be any improvement in expected squared error v. However, the uncorrelated models ,i.e., $\mathbb{E}\left[\delta_i \delta_j\right] = 0$ would shrink the expected squared error by k times. Thus, the proposed dynamic model would perform significantly better than the individual models with approximately half the error. In addition, the speed wont degrade much due to closed-form solution of the temporal weights, which can be computed in $\mathcal{O}(1)$ time complexity.

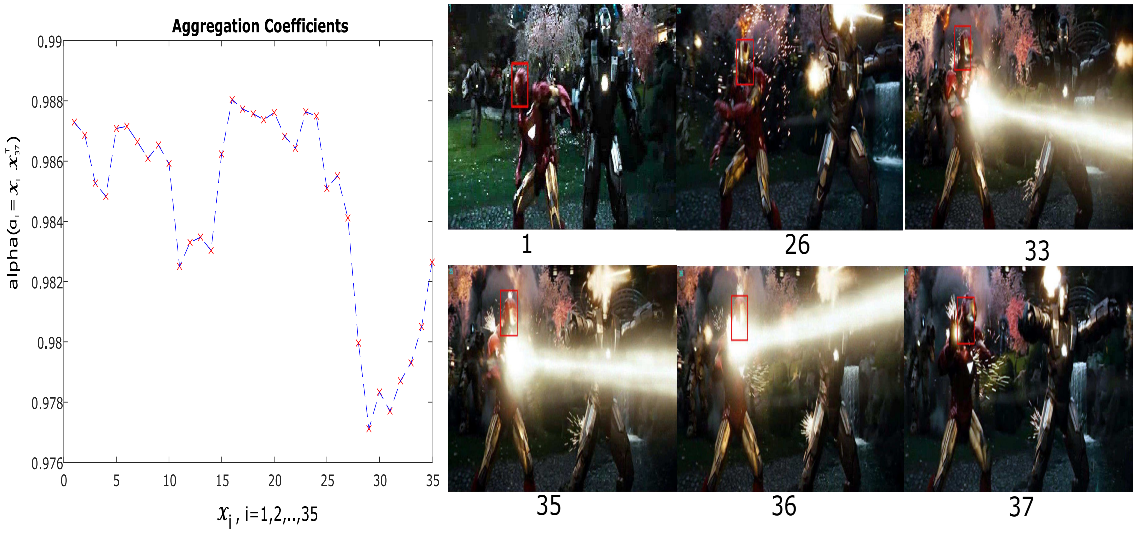

Figure 3.15: Coefficients of aggregation α, which are used to modulate the preceding features of the corresponding frames based on similarity rational. Here, x_{37} has been projected onto $X(1:35)$, where $n = 3140, m = 37$, i.e., $x_i \in \mathbb{R}^{1\times 3140}, i = 1, 2, \ldots, m$, $X \in \mathbb{R}^{37\times 3140}$, $Y \in \mathbb{R}^{37\times 4}$, and $\omega \in \mathbb{R}^{3140\times 4}$.

3.4.4 Revisiting Temporal Regression

Here, we discuss about some key technical contributions in the proposed framework that offers additional gain in the performance standard. One of the major observations is that elimination of immediate past frame ($(m-1)^{\text{th}}$) from computation of temporal coefficients provides improvement over inclusion of the immediate past frame. We remark that the $(m-1)^{th}$ frame may not necessarily correlate maximally with the current frame. Also, the output of the tracker may sometimes lead to false positive bounding box which will incrementally allow it to drift away from the actual target. In other words, the trajectory of an object, moving in a straight line may become curved during regression due to the $(m-1)^{th}$ false positive localization. A possible solution could be to eliminate few past frames from TR, but this would restrain the learning of recent appearance changes. Therefore, a possible solution is to remove the effect of last frame from training, which would capture the actual straight line trajectory, and thus, it will assist in few scenarios where drastic change is a major concern. We have eliminated the experiments with removal of more immediate frames based on qualitative analysis, and showcase the efficacy of removing immediate past frame on whole OTB50 dataset. However, this approach may become troublesome when the actual trajectory has abrupt deviation from previous estimates. So, the weighted mean ensemble of detector, which is mostly right (more weightage, 0.7), and TR would be useful to tackle this issue. Figure 3.15 illustrates that the current frame $m = 37$ has relatively lower correlation with immediate past frames than long term frames. A few sample frames from Ironman sequence of OTB50 are also shown to visually validate our hypothesis.

In order to reduce the computational complexity for very large sequences, we have borrowed few ideas on Long Short Term Memory (LSTM) from Recurrent Neural Networks (RNN), and used only previous 50 frames excluding the immediate past frame. For ease of understanding, we have explained the fundamental principles of our architecture in Section 3.4. However, the actual implementation has been refined slightly based on the aforementioned key observations, which we have explained in Algorithm 1 for reproducibility.

Algorithm 1 Temporal Regression

```
Define global cell arrays $X = \{\}$ and $Y = \{\}$
repeat at each frame
    Input: (ECO) detector output $y_m$, and feature vector $x_m$.
    Processing: Transform $x_m$ into PCA space with $90\%$ captured variance, $x_m \leftarrow$PCA$(x_m)$. Concatenate $y_m$ with $x_m$, $x_m(end + 1 : end + 4) = y_m$. Accumulate new $x_m \in \mathbb{R}^{1\times n}$ in $X$, $X\left\{end + 1, 1\right\} = x_m$ and $y_m \in \mathbb{R}^{1\times 4}$ in $Y$, $Y\left\{end + 1, 1\right\} = y_m$. Let frame($m$) represent the number of elements in cell X, which is equivalent to the current frame index. Initially, $m = 1$ and $l = 2$.
    if $m > l$ then
        Assign $s$ to $\max(m - 50, 1)$.
        Compute aggregation coefficients, $\alpha = \frac{X(s:m-l)*X(m)^T}{n}$.
        Smoothen using sigmoid activation, $\alpha = sigmoid(\alpha)$.
        Employ weighted Aggregation, $\widetilde{X} = X. * \alpha$.
        for $i = 1$ to $n$ do
            Select $i^{th}$ feature from all frames, $f_i = \widetilde{X}(:, i)$.
            Estimate Shannon Entropy of $f_i$, $H(f_i) = -\mathbb{E}_{f_i \sim \mathbb{P}_{f_i}}\left[\log \mathbb{P}_{f_i}(\mathrm{f})\right]$.
            Update features, $\widetilde{f_i} = f_i * H(f_i)$, and $\widetilde{X}(:, i) = \widetilde{f_i}$.
        end for
        Consider $\widetilde{X} = \widetilde{X}(s : m - l)$, and $\widetilde{Y} = Y(s : m - l)$.
        Compute temporal regression parameters, $\omega = \left[\widetilde{X}^T\widetilde{X} + \lambda\xi^T\xi\right]^{-1}\widetilde{X}^T\widetilde{Y}$.
        Generate predictions, $\widetilde{y_m} = x_m\omega$.
        Ensemble detector and regression outputs, $op$ = $\frac{0.7*y_m+0.3*\widetilde{y_m}}{2}$.
        Update target attributes in $x_m$, $x_m(n - 3 : n) \leftarrow \frac{y_m+\widetilde{y_m}}{2}$.
    end if
    Process the next frame, $m = m + 1$
    Output: $op$
until the end of frames.
```

Chapter 4

Tracking Benchmarks and Evaluation Metrics

Datasets play a critical role in almost all computer vision tasks. In the case of the object classification problem, there has been a tremendous evolution from Caltech101 [77] to PASCAL VOC [78] and then to large-scale ImageNet [79]. While such an evolution has also occurred in the case of tracking, it has been at smaller scale and a slower pace, and has its fair share of issues. Most video sequences in initial datasets were recorded in an unnatural experimental environment, or in some cases selected to highlight the advantages of the proposed tracker. Furthermore, they lack a common protocol for ground truth annotation, and are typically small in number. These issues are being addressed by recent datasets and benchmarks [14, 80, 5]. In this section, we first briefly review publicly available model-free tracking datasets, and then introduce the datasets and corresponding evaluation methods used in this thesis.

4.1 Overview of Tracking Datasets

4.1.1 Amsterdam library of ordinary videos (ALOV++) dataset

Generality is a very important feature for good model-free trackers. Smeulders *et al.*[80] argued that most trackers, have only been evaluated on a limited number of sequences, and the evaluation results are insufficient to make conclusive remarks on the validity and robustness of the proposed methods in a variety of circumstances. To address this issue, a large and diverse dataset was proposed. The ALOV++ dataset contains 315 video sequences with 89,364 frames in total from several sources: 22 sequences come from standard and recent tracking datasets, 65 sequences are from performance evaluation of tracking and surveillance (PETS) workshop [81], and 250 new sequences are collected from YouTube with 64 different types of targets. The ALOV++ dataset is annotated with a regular bounding box (i.e., axis-aligned box) enclosing the target. Due to the large size of the dataset, ground truth is manually annotated every fifth frame, while the annotation of the intermediate frames is obtained by linear interpolation. ALOV++ is available online at `http://www.alov300.org`.

4.1.2 NUS people and rigid objects (NUS-PRO) dataset

It is the largest publicly available tracking dataset so far, and contains 365 video sequences collected from YouTube. All the sequences in NUS-PRO belong to five categories, namely, face, pedestrian, sportsman, rigid object and long sequences. The five categories contain 17 kinds of objects in all. Many video sequences in the NUS-PRO dataset are recorded by hand-held cameras which makes it close to real-life scenarios, e.g., videos contain abrupt object movement or motion blur. Moreover, occlusion, usually missing or casually marked in other tracking datasets, is elaborately considered and annotated in three categories: no occlusion, partial occlusion and full occlusion. The NUS-PRO dataset and the evaluation system are available at `http://www.lv-nus.org/pro/nus_pro.html`.

4.1.3 Princeton tracking benchmark (PTB) dataset

Song and Xiao [82] constructed an RGBD tracking dataset of 100 video sequences, which are captured with a standard Microsoft Kinect 1.0. It is the first attempt to build a tracking dataset with depth information, which significantly reduces the ambiguity existing in RGB images [83], and can be used to prevent model drifting and handle occlusion cases. However, due to the constraint of the recording device, the depth of the captured object can only vary from 0.5 to 10 meters, and thus all the RGBD video sequences are captured indoors. The PTB dataset and the evaluation system are available at `http://tracking.cs.princeton.edu`.

We use the-state-of-the-art object tracking benchmark (OTB) [14] and the visual object tracking (VOT) challenge [5] datasets extensively in this thesis, and will discuss them in 4.2. A summary of tracking datasets is shown in Table 4.1.

Dataset	#Videos	Groundtruth (rectangle)	Data
TB-50/100 [14]	100	Axis-aligned	RGB,gray
VOT [5]	60	Rotated	RGB
ALOV++ [80]	315	Axis-aligned	RGB,gray
PTB [82]	100	Axis-aligned	RGBD
NUS-PRO [77]	365	Axis-aligned	RGB

Table 4.1
A summary of popular tracking datasets.

4.2 Datasets and evaluation methods used

4.2.1 Object tracking benchmark (OTB) dataset

The object tracking benchmark dataset [14], named OTB, is a collection of 50 commonly used tracking sequences, where the object varies in scale, has fast motion, or is occluded. The first frame of each sequence in OTB is illustrated in Figure 4.1.

In order to present the progress of tracking algorithms and set a general benchmark, 29 methods are compared in [14]. Two well-adopted evaluation methodologies are used: precision and success. Precision reflects the center location error. It is measured as the percentage of frames whose predicted object location (center of the predicted box) is within a distance varying between 0 and 50 pixels from the center of the ground truth box. The precision score is the percentage value when threshold distance is set to 20 pixels. The success measure is based on the bounding box overlap. It shows the percentage of frames whose intersection over union overlap with the ground truth annotation is over a threshold, varying between 0 and 1. Instead of using a fixed threshold, the area under curve (AUC) of the success plot determines the success score in order to rank the algorithms. The robustness of different trackers is evaluated with the following three procedures.

- **One-pass evaluation (OPE):** It is a conventional method to evaluate trackers. All the trackers are run on the test sequences with the initializations from the ground truth position in the first frame, and the average precision and success scores are measured.

- **Temporal robustness evaluation (TRE):** Each sequence for testing is divided uni-

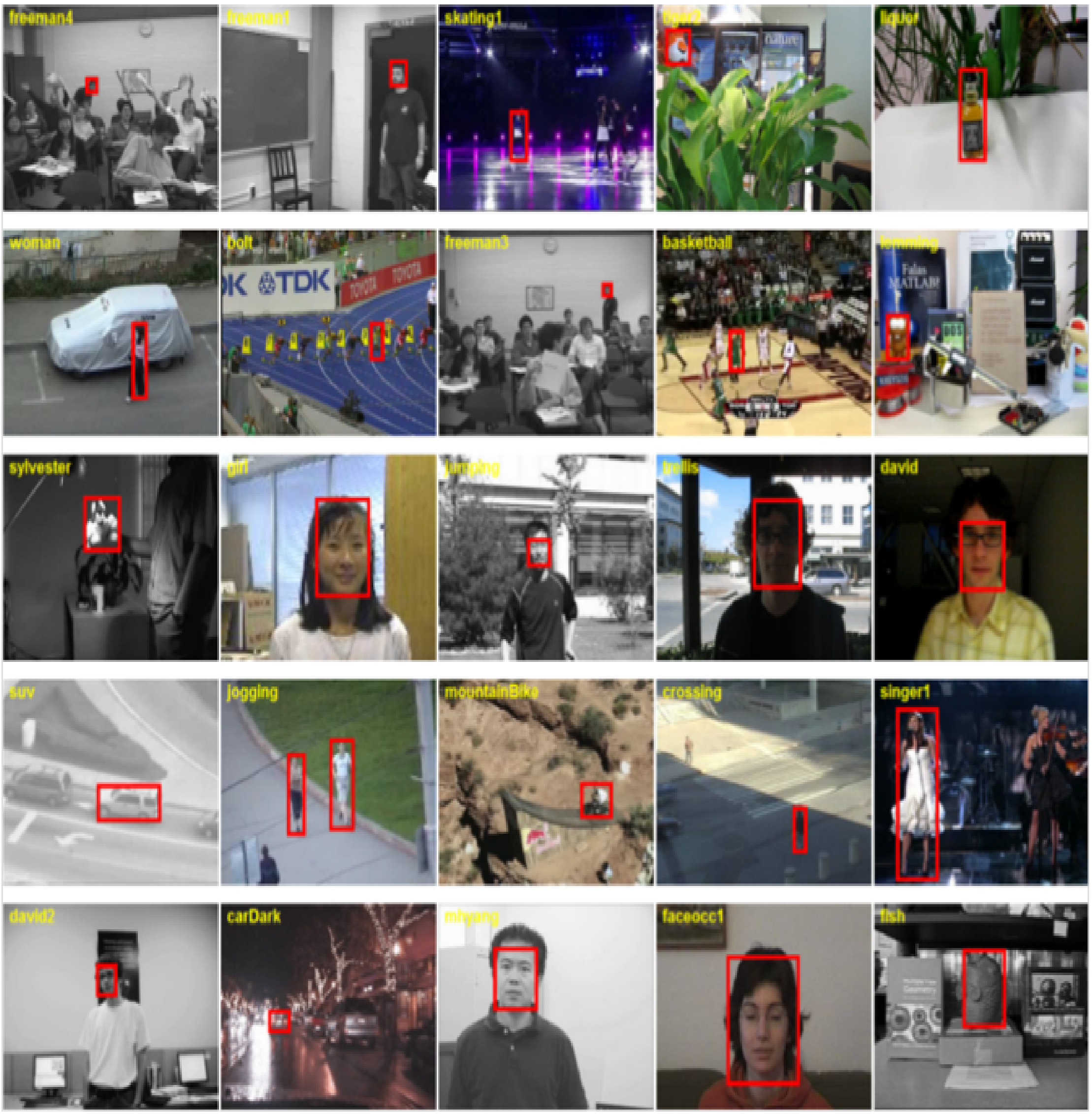

Figure 4.1: Visualizing initialization frame of few sequences from OTB [14] benchmark.

formly into 20 segments. Each tracker is initialized at the beginning of a segment and evaluated until the end of the entire sequence. The tracking results of all the 20 tests are averaged to generate the precision and success scores.

- **Spatial robustness evaluation (SRE):** For every sequence, each tracker is initialized in the first frame with shifted or scaled ground truth bounding box. As a default, each tracker is evaluated 12 times with different initial bounding box settings: eight spatial shifts including four center shifts and four corner shifts (10% of target size), and 4 scale variations (i.e., 0.8, 0.9, 1.1 and 1.2) with respect to the ground truth in the first frame. The precision and success scores are calculated from the average of all these 12 evaluations in order to rank the trackers.

4.2.2 Visual object tracking (VOT) challenge dataset

The visual object tracking (VOT) challenge was introduced in 2013 with the aim of providing a standardized platform to evaluate singlecamera, single-target, model-free, causal short-term tracking algorithms. It has been organized as an annual workshop in conjunction with ICCV or ECCV conferences. In each workshop, a fully annotated dataset with several per-frame visual attributes is released. Each frame in the dataset is manually or semi-automatically labeled with six visual attributes, including occlusion, illumination change, motion change, size change, camera motion and unassigned. In addition to the dataset, an evaluation toolkit is also developed and actively maintained, which allows easy integration of third-party trackers for fair comparison. We use the datasets released for the challenge organised in 2016, namely VOT2016 [5] benchmark.

The evaluation scheme of VOT challenge uses accuracy and robustness measures to compare trackers, due to their high level of interpretability [84]. Raw accuracy is computed as the mean intersection over union score with the ground truth bounding box over the entire sequence (while discarding ten frames immediately following a tracking failure to further reduce the bias in accuracy measure), and raw robustness is the number of times the tracker has failed. A tracking failure is signaled in a frame t if the predicted box does not overlap with the ground truth annotation. In this case, the tracker is restarted from scratch in frame $t + 5$ with the corresponding ground truth annotation in order to alleviate the bias in robustness measure. For a robust comparison, the scores are averaged over 3 repitive runs of the tracker to account for any stochastic behavior. The first frame of each sequence in VOT2016 is illustrated in Figure 4.2.

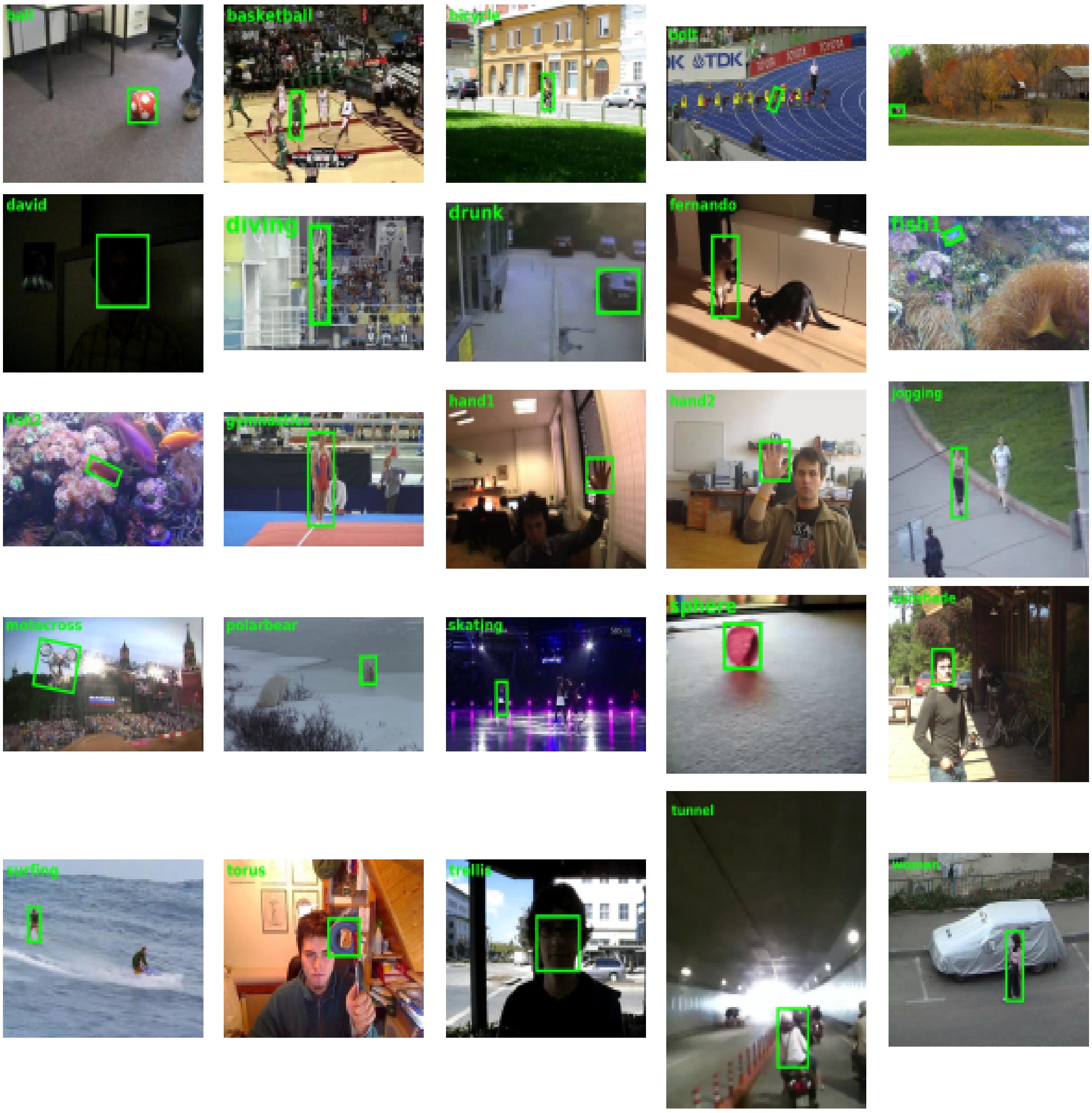

Figure 4.2: Visualizing initialization frame of few sequences from VOT [5] benchmark.

Chapter 5

Experimental Details and Analysis

5.1 Introduction

Here, we provide necessary details regarding our experimental setup, and critically analyse our contributions along with other state-of-the-art trackers. At first, we provide the experimental details of deep learning based trackers (Section 5.2). Thereafter, the performance assessment of correlation filter based trackers (Section 5.3), followed by the essential evaluation results on temporal regression models (Section 5.4).

5.2 Experiments in Deep Learning based Trackers

We have evaluated original CFnet and our modified CFnet on 43 sequences out of OTB50 with 3 repetitions for each sequence in order to get a rough estimation of the performance. These particular sequences were selected based on the toughness of deformations incurred in the object of interest. For the evaluation on 43 sequences, we have used OTB-TRE function which has been provided in original CFnet [18] codes repository. The tracking performance varies from machine to machine based on whether GPU support is enabled. The reason for this is the numerical effects which gets accumulated over time. Due to which, most of the trackers suffer from a slight variation in results when re-evaluated on different machines. In order to avoid this effect, we have evaluated the original tracker and all our modifications under exactly same circumstances i.e. same sequences, same system and same evaluation function. The results are given in Table 5.1. From Table 5.1, it is clearly observed that our proposed displacement 3.2.3 and scale 3.2.4 correction schemes have improved the performance of original CFnet [18]. The success and precision values for different angles of rotation are shown in Table 5.1. From Table 5.1, it is clear that the tracker achieves optimal performance for the angle of rotation $\zeta = 8^{\circ}$ and its performance

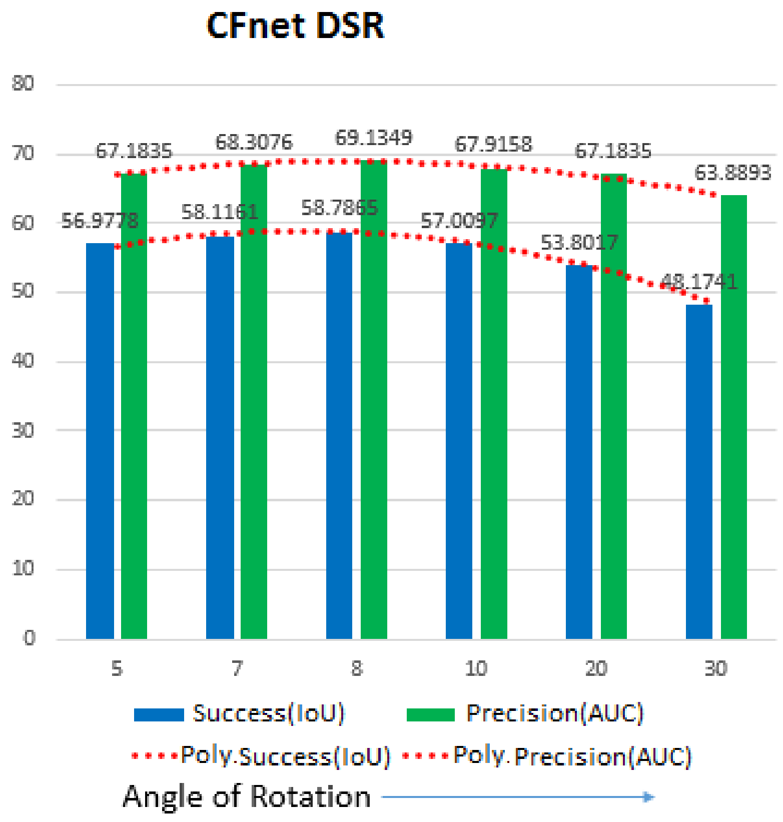

Figure 5.1: A rough estimation for optimal angle of rotation ζ using OTB-TRE function given in CFnet [18] code repository. A chart of success(AUC) and precision(Threshold) vs various angle of rotation.

decreases with increase in angle of rotation. This is evident from practical point of view as there won't be drastic change in orientation between two subsequent frames.

Final evaluation is done using OTB toolkit for all 50 sequences [14] and only important results are shown due to space constraint. The success rate of original CFnet could not be plotted because of the unavailability of final bounding boxes in OTB results database during the time of writing the thesis. A comparison between our modified CFnet and current state-of-the-art trackers is shown in Figure 3.9. Due to lack of rotated bounding box results of other trackers, we had to use axis aligned bounding box during accuracy assessment. So there is a slight improvement in accuracy and precision as shown in Table 5.2. This performance will certainly be enhanced if compared with rotated bounding box results which unfortunately isn't available for most of the trackers. The success(AUC) and precision plots obtained by using fully integrated OTB toolkit are shown in Figure

Tracker	Success(AUC)	Precision (Threshold)
CFnet	57.3447	65.0869
CFnet D	59.0531	69.3120
CFnet DS	59.0531	70.3673
CFnet DSR ($\zeta = 8°$)	58.7865	69.1349

Table 5.1
Integration of Displacement 3.2.3, Scale 3.2.4 Correction and rotation invariant strategy 3.2.5. DSR stands for Displacement correction, Scale correction and Rotation invariant strategy respectively.The results are obtained using the OTB-TRE evaluation function provided in original CFnet [18] code. This evaluation is similar to OTB toolkit evaluation, but not exact. The reason behind using this evaluation function is to make a fair comparison with the original CFnet [18]. The comparison is done for axis aligned bounding box results.

Tracker	Success(AUC)	Precision (Threshold)
CFnet	52.7	70.2
CFnet DS	52.9	70.0
CFnet DSR ($\zeta = 8°$)	52.8	71.5

Table 5.2
Fully integrated OTB-OPE comparison. The results are obtained using OTB toolkit.

5.2 and Figure 5.3 respectively.

As per the results obtained using fully integrated vot-toolkit as shown in Table 5.3, an improvement of 15.57% in accuracy rank and 14.3% in robustness rank have been observed with no degradation in overlap ratio. Since the absolute value of the accuracy and robustness rank varies based on the trackers used for evaluation, the relative improvements of proposed methods over the original have been used to showcase the efficiency. The ranking plot for experiment baseline is shown in Figure 5.4, which depicts the improvement in accuracy and robustness of proposed modified Siamese DSR over the original SiameseFC.

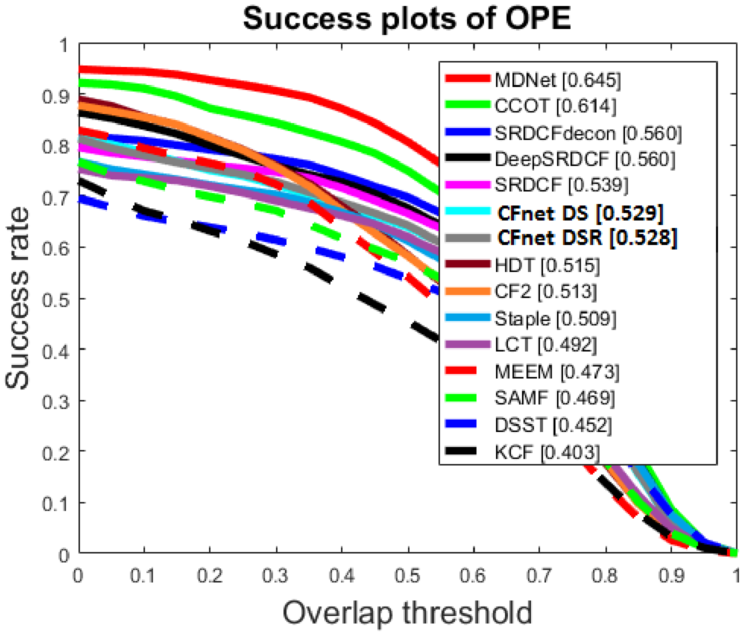

Figure 5.2: OTB50 success plot(AUC) obtained using OTB toolkit (Values are scaled down from 100 to 1 by the toolkit). Original CFnet-conv2 success rate(OPE) for OTB50 is equal to 0.527 as per the evaluation in our system.

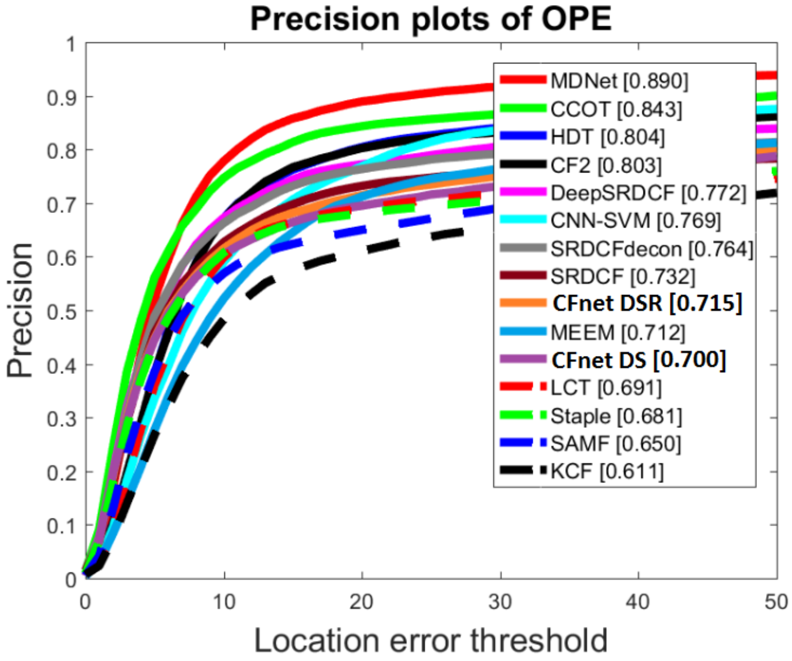

Figure 5.3: OTB50 precision plot obtained using OTB toolkit. Original CFnet-conv2 precision(OPE) for OTB50 is equal to 0.702 [18] as per the evaluation in our system.

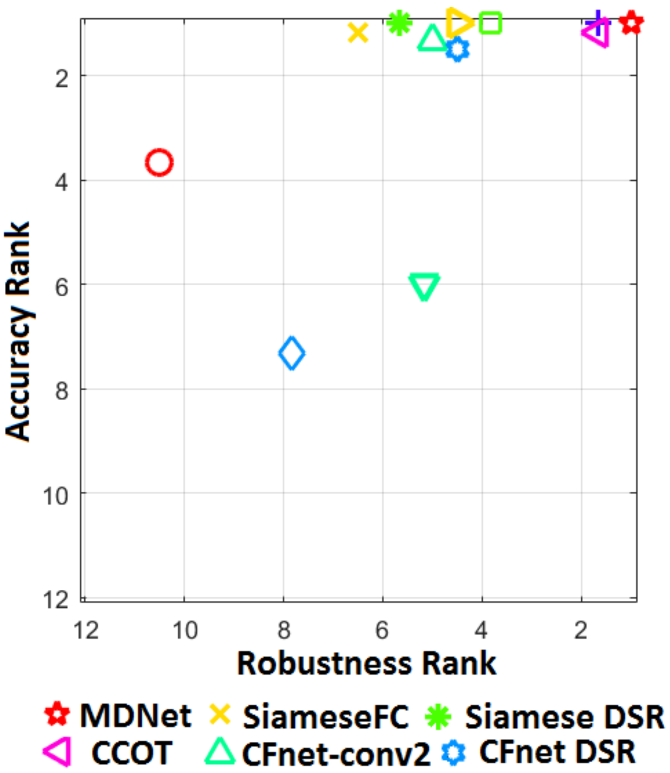

Figure 5.4: VOT Ranking Plot for experiment baseline(mean).

Tracker	Accuracy Rank	Robustness Rank	Mean Overlap
MDNet	1.00	1.00	0.57
CCOT	1.17	1.67	0.54
SiameseFC	1.17	6.50	0.52
Siamese DSR	1.00	5.67	0.52
CFnet-conv2	1.33	5.00	0.52
CFnet DSR	1.50	4.50	0.52

Table 5.3
VOT AR ranking for experiment baseline. Siamese DSR represents the proposed modified network over original SiameseFC[9]. CFnet DSR represents the proposed modified network over actual CFnet-conv2 [18]

Tracker	ECO	D-ECO	DF-ECO	R-ECO	RF-ECO	RD-ECO	RDF-ECO	RIDF-ECO
AEO	0.357	0.360	0.362	0.383	0.386	0.395	0.402	0.433
%Gain	Baseline	0.8	1.4	7.3	8.1	10.6	12.6	21.3

Table 5.4
Quantitative evaluation of individual components on a set of 16 challenging videos.

5.3 Experiments in Correlation Filter based Trackers

In order to perform an unbiased analysis that may arise due to varying numerical precision of different systems, we evaluate all the experiments, including baseline SRDCF and ECO on the same system under identical experimental setup. We use the similar parameter settings as baseline, apart from the additional parameters $\delta = 5^\circ$, and $A = 2$ in rotation adaptive filters. In IC, we use output intensity range $[0, 255]$ for contrast stretching, and a threshold 0.5 for unsharp masking. A detailed description on the use of these parameters in contrast stretching and unsharp masking can be found in [75]. We use VOT toolkit [5] for evaluating the performance of compared trackers on VOT2016 dataset.

We progressively integrate Displacement consistency (D), False positive elimination (F), Rotation adaptiveness (R), Illumination correction (I), and their combinations into ECO framework for faster experimentaion, and assimilate the impact of each individual component on AEO, which is the standard metric on VOT benchmark. To analyze the ability of illumination and rotation adaptiveness separately, we evaluate each individual component on a set of 16 videos (Table 5.4). The set is constructed from the pool of VOT2016 dataset. A video is selected if its frames are labelled as either severe deformation, rotation, or illumination change by the VOT2016 benchmark. Note that, the FPE scheme improves the performance in every integration, and illumination correction provides a gain of 7.7% over base RDF-ECO. As per the results in Table 5.4, the proposed ideas independently and together provide a good improvement relative to base model.

We further evaluate the top performing models, including individual components of SRDCF, on whole VOT2016 dataset (Table 5.5). As per the results in Table 5.5, the I-SRDCF, RDF-SRDCF, and RIDF-SRDCF provide a considerable improvement of 3.53%, 10.60%, and 11.41% in AEO, 4.83%, 17.87%, 13.04% in robustness, respectively. The

Trackers	SRDCF	I-SRDCF	RDF-SRDCF	RIDF-SRDCF	TCNN	CCOT	ECO	MDNet	RIDF-ECO
AEO	0.1981	0.2051	0.2191	0.2207	0.3249	0.3310	0.3563	0.3584	0.3624
Failure Rate (Robustness)	2.07	1.97	1.70	1.80	0.96	0.83	0.78	0.76	0.73

Table 5.5

State-of-the-art comparison of proposed methods on whole VOT2016 dataset.

RIDF-ECO performs favourably against the state-of-the-art trackers with a slight improvement of 5.27% in AEO, and as high as 6.41% in robustness. Note that, the percentage improvements are computed relative to baseline. In Figure 5.5, we show the oriented bounding box results which are obtained from the RIDF-SRDCF tracker. Due to unavailability of results on whole VOT2016 dataset in the required format, we were unable to compare few rotation adaptive trackers [62, 63, 35] with our trackers. However, since our rotation adaptive tracker: RIDF-ECO offers significant gain relative to baseline (ECO) that outperforms few state-of-the-art rotation adaptive trackers [62, 63, 35], we report that the proposed rotation adaptive scheme will surpass these counter parts with ease. The overall time complexity of our RIDF-SRDCF tracker sums up to

$$\mathcal{O}\left(ASdMN\log MN + (ASdMN + ASMN)\,N_{Ne} + MN + \left(d+k^2\right)dMNN_{GS}\right), \tag{5.1}$$

excluding the feature extraction, where K, N_{Ne}, and N_{GS} denote the number of non-zero Fourier coefficients in $\hat{w}$, the number of iterations in sub-grid detection, and the number of iterations in Gauss-Seidel optimization, respectively. Note that, the last term dominates the overall computational complexity.

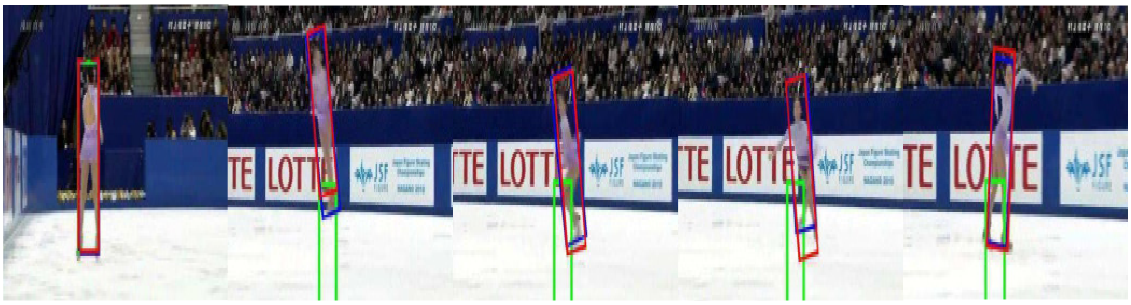

Figure 5.5: Sample frames from the sequence iceskater1 of VOT2016. The blue, green, and red rectangle shows the groundtruth, SRDCF, and RIDF-SRDCF outputs, respectively. The RA correlation filters efficiently detect the orientation of the traget object.

5.3.1 Computational Complexity of Rotation Adaptiveness

The Fast Fourier Transform (FFT) of a 2-dimensional signal of size $M \times N$ can be computed in $\mathcal{O}(MN \log MN)$. Since there are d feature layers, S scales, and $(2A+1)$ orientations, the training and detection stage of our algorithm requires $\mathcal{O}\left(ASdMN \log MN\right)$ FFT computations. To compute the convolution response, the computed FFTs require $\mathcal{O}\left(ASdMN\right)$ multiplication operations, and $\mathcal{O}\left(ASMN\right)$ division operations. The division operations are used in False Positive Elimination (FPE) strategy. Assuming that the Newton's optimization converges in N_{Ne} iterations, the total time complexity of matrix multiplication and FPE sums up to $\mathcal{O}\left(\left(ASdMN + ASMN\right) N_{Ne}\right)$. In contrast to standard SRDCF [7], we learn the multi-resolution filter coefficients from properly oriented training samples. After detection of orientation through optimization of total energy content on a coarse grid, the training samples are oriented appropriately in $\mathcal{O}\left(MN\right)$ time complexity. The fraction of non-zero elements in A_t of size $dMN \times dMN$, as given in standard SRDCF, is bounded by the upper limit $\frac{2d+k^2}{dMN}$. Thus, the total time complexity of standard SRDCF training, assuming that the Gauss-Seidel optimization coverges in N_{GS} iterations, sums up to $\mathcal{O}\left(\left(d+k^2\right) dMNN_{GS}\right)$. In addition to the standard SRDCF training, our approach requires $\mathcal{O}\left(MN\right)$ operations to orient the samples, leading to a total complexity of $\mathcal{O}\left(MN + \left(d+k^2\right) dMNN_{GS}\right)$. Therefore, the overall time complexity of our RIDF-SRDCF is given by,

$$\mathcal{O}\left(ASdMN \log MN + \left(ASdMN + ASMN\right) N_{Ne} + MN + \left(d+k^2\right) dMNN_{GS}\right). \tag{5.2}$$

Note that the overall complexity is largely dominated by $\mathcal{O}\left(\left(d+k^2\right) dMNN_{GS}\right)$, leading to slight increment in computational cost, but significant improvement in overall performance of RIDF-SRDCF relative to standard SRDCF.

5.3.2 Detailed Experimental Evaluations

Here, we demonstrate additional evaluation results to experimentally validate the efficacy of our contributions in visual object tracking. In Section 5.3.2.1 and 5.3.2.4, we show the results of RIDF-SRDCF and RIDF-ECO, respectively.

5.3.2.1 Evaluation of RIDF-SRDCF

In this section, we show the experimental results of RIDF-SRDCF evaluated by fully integrated vot-toolkit on whole VOT2016 dataset [5]. To qualitatively assess the overall performance of RIDF-SRDCF, we compare our results with baseline approach on few challenging sequences from VOT2016 dataset, as shown in Figure 5.6. Further, we quantitatively assess the performance by comparing the Average Expected Overlap (AEO) of few correlation filter based trackers, as shown in Figure 5.7.

5.3.2.2 Qualitative Analysis

Figure 5.6 shows the qualitative comparison of the proposed RIDF-SRDCF tracker with various correlation filter based trackers.

5.3.2.3 Quantitative Analysis

The proposed RIDF-SRDCF outperforms the standard SRDCF in most of the individual categories that leads to 11.4% and 13.04% overall improvement in AEO and robustness, respectively. The categorical comparison, as can be inferred from Figure 5.7, shows 56.25%, 23.53%, 38.46%, 5.26%, and 16.66% gain in Illumination change, Size change, Motion Change, Camera motion, and Empty categories, respectively. Note that the percentage improvement is computed relative to base SRDCF.

5.3.2.4 Evaluation of RIDF-ECO

To assess the overall performance of the proposed RIDF-ECO, we evaluate the tracker on whole VOT2016 dataset [5]. The qualitative and quantitative analysis of RIDF-ECO along with few state-of-the-art trackers are shown in Figure 5.8, and Figure 5.9, respectively.

5.3.2.5 Qualitative Analysis

Figure 5.8 shows the qualitative comparison of the proposed RIDF-ECO tracker with state-of-the-art trackers.

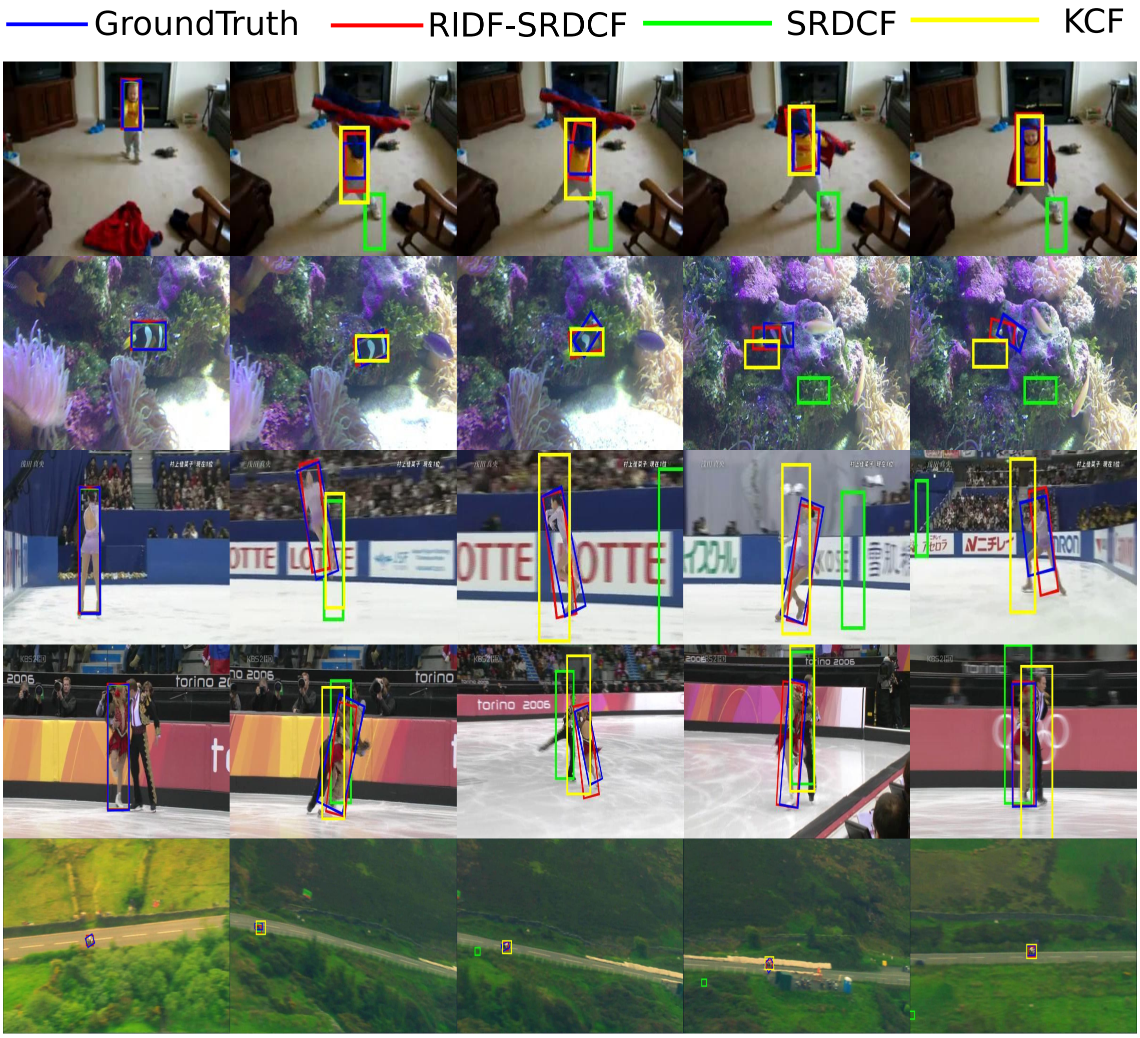

Figure 5.6: Qualitative analysis of RIDF-SRDCF compared with baseline SRDCF and few other correlation filter trackers. The proposed tracker performs favourably against the other correlation filter trackers. The rotation adaptive filters assist in determining the orientation of the target object effectively that leads to substantial gain in overall performance.

5.3.2.6 Quantitative Analysis

The proposed RIDF-ECO outperforms the state-of-the-art trackers in most of the individual categories that leads to 1.72% and 18.5% overall improvement in AEO and robustness, respectively. Though the performance of RIDF-ECO in individual categories is reasonably good, except Camera Motion, the gain is significant in Occlusion (23.81%), Motion Change (13.33%), and Empty (13.33%) categories. The improvement in Illumination change category is not as significant as in RIDF-SRDCF because RDF-ECO performs reasonably well in this category, leaving slightest scope for substantial gain in RIDF-

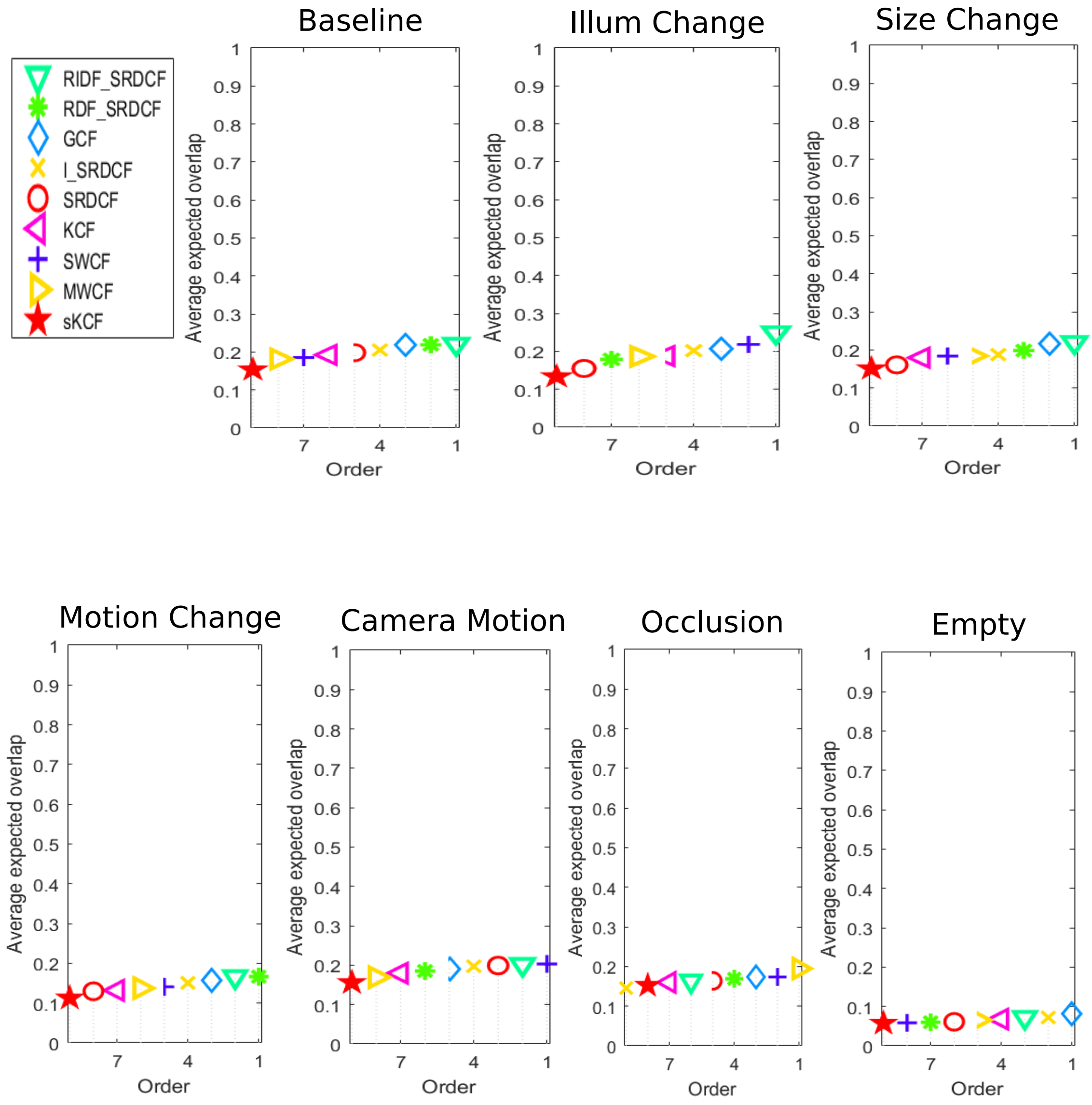

Figure 5.7: Average Expected Overlap analysis of correlation filter based trackers.

ECO. Also, we observe that the RIDF-ECO degrades the performance of base ECO [38]. Moreover, the overall performance of RIDF-ECO is reasonably better than the base ECO in terms of detecting orientation, eleminating false positives, reducing failure rate etc. that leads to 1.71%, and 6.41% gain in Average Expected Overlap (AEO), and robustness, respectively.

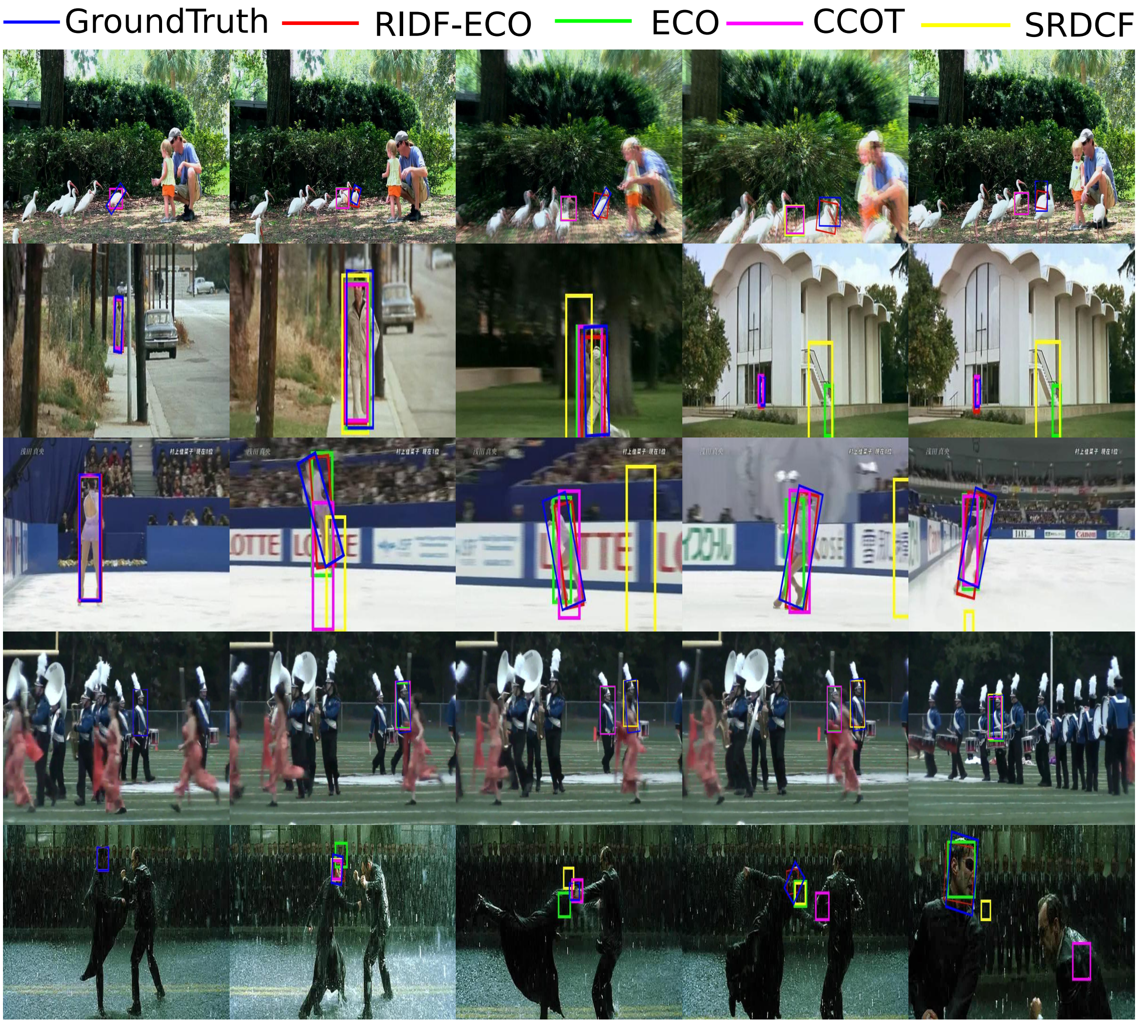

Figure 5.8: Qualitative analysis of RIDF-ECO compared with baseline ECO and few other state-of-the-art trackers. The proposed tracker performs favourably against the state-of-the-art trackers. The orientation of the target object is determined efficiently to some extent, mainly due to rotation adaptive correlation filters.

5.4 Experiments in Temporal Regression based Trackers

Here, we detail our experiments and draw essential inferences to validate our methodology. We analyse our approach on two popular object tracking benchmarks: OTB50 and VOT2016 under various critical circumstances. To avoid the ambiguity caused by numerical computation of different machines, we evaluate both the baseline and our proposed trackers on the same machine with exactly same experimental setup. We use the parameter settings of ECO [38], including feature extraction, factorized convolution and optimization, for generating detector output. We develop our algorithm by progressively integrating our contributions into baseline. We demonstrate the impact of individual components

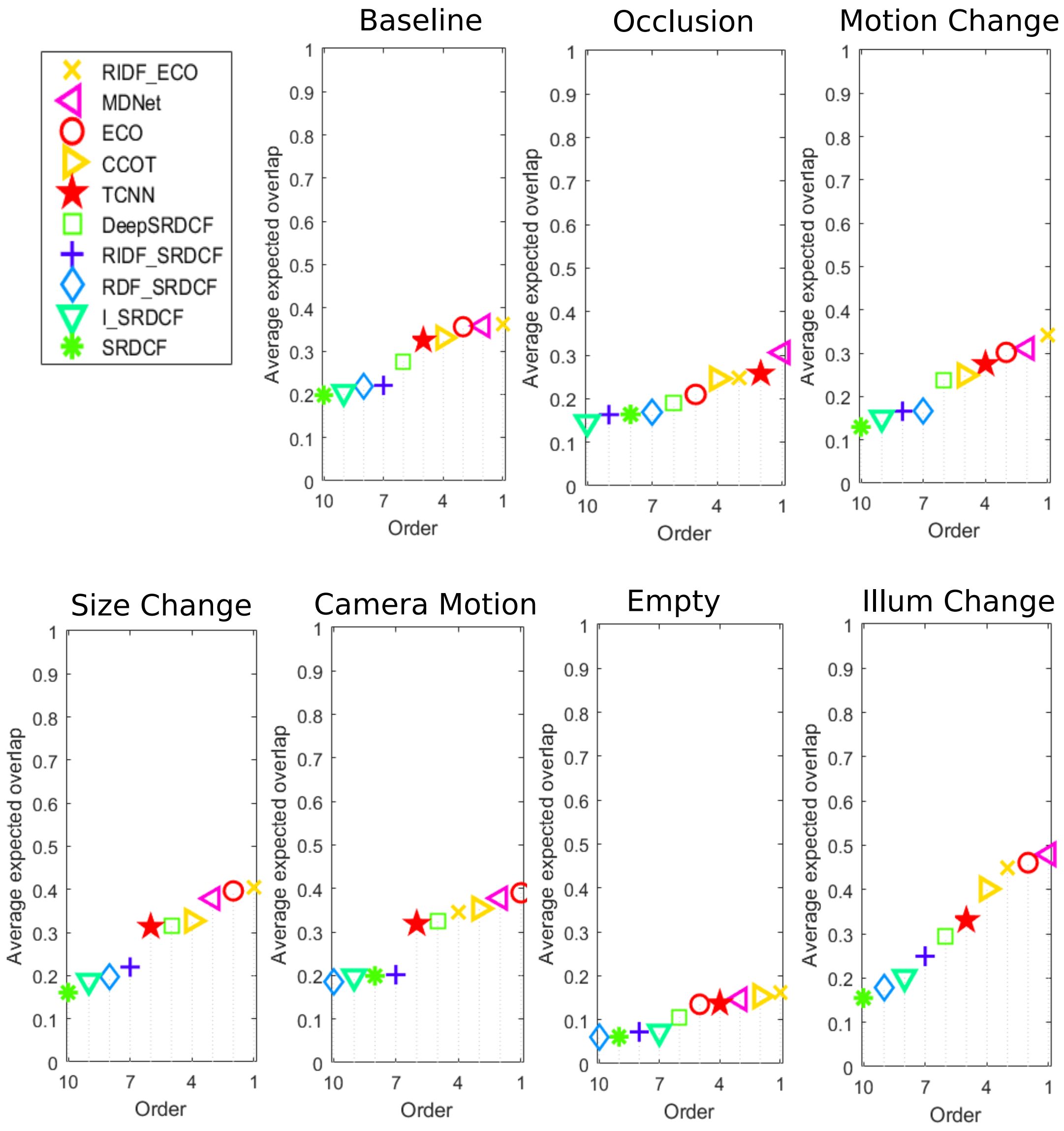

Figure 5.9: Average Expected Overlap analysis of RIDF-ECO and state-of-the-art trackers.

Tracker	WAEF	TREF	ECO	TR2	TR1	WAEF1	ECO_EF	WAEF2
Success Rate	0.651	0.648	0.643	0.627	0.619	0.615	0.611	0.610
Precision	0.880	0.877	0.874	0.849	0.839	0.825	0.822	0.814

Table 5.6
The success and precision area under the curve (AUC) of the individual components of our proposed framework on OTB50.

by performing extensive experimentation on OTB50. We compare our top-performing trackers with state-of-the-art trackers and show compelling results in all the challenging categories of OTB50. Figure 5.10 shows the qualitative analysis of the proposed framework.

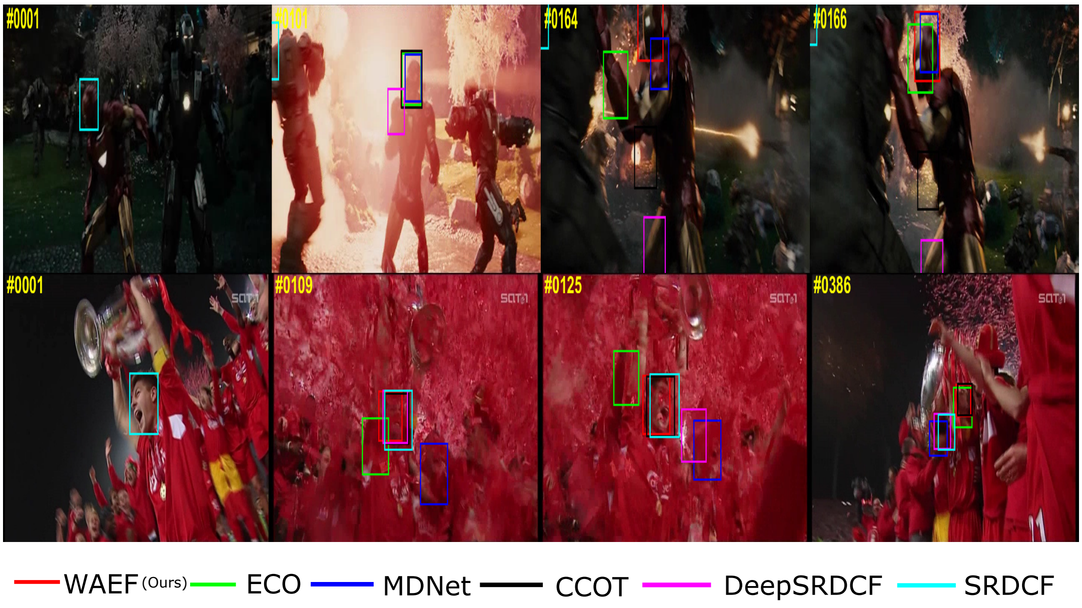

Figure 5.10: Qualitative anaysis of our WAEF tracker and several other state-of-the-art trackers on two of the toughest sequences: Ironman and Soccer from OTB50 dataset. Our WAEF tracker performs favourably against the top trackers.

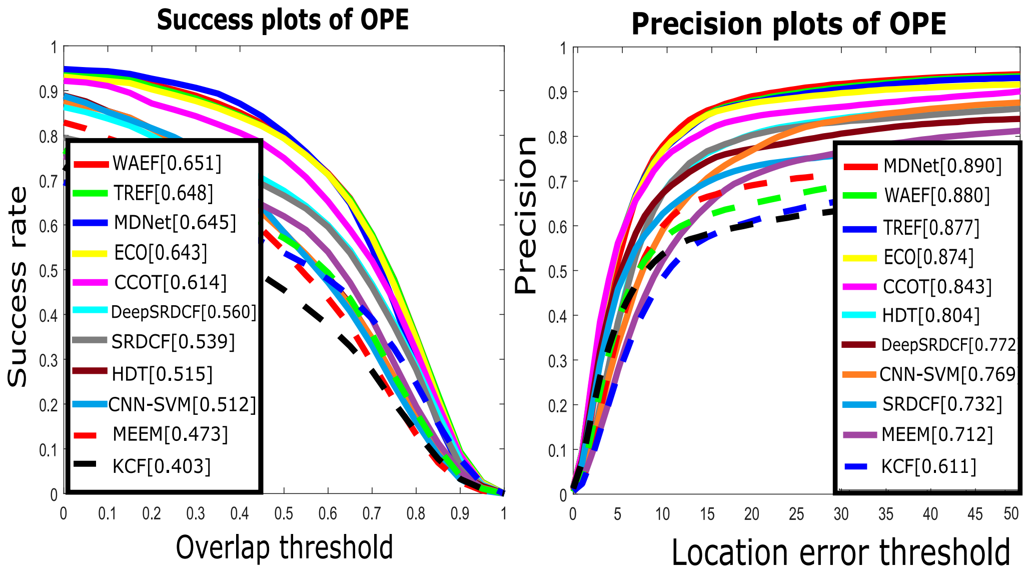

Figure 5.11: The success and precision plots of our proposed WAEF, TREF, and several state-of-the-art trackers on OTB50 dataset.

In Table 5.6, we analyse the performance of each method separately. To our surprise, the enhancement filter (ECO_EF) degrades the performance on OTB50, even though it offers appealing results on VOT2016 with 1.48% improvement in AEO (Table 5.8). TR1 and TR2 denote the temporal regression with training features from $\max(m-50, 1)$ to $m-1$ and $m-2$, respectively. Note that the TR1 and TR2 do not use weighted aggregation while computing ω. It is evident that TR2 is better than TR1 both in accuracy and robustness, which validates our hypothesis of excluding immediate previous frame from training TR model. Despite the weak performance of TR and ECO_EF compared to baseline, their composition tracker TREF outperforms the baseline in Success rate and Precision. Further, the WA and TREF consolidate into Weighted Aggregation with Enhancement Filter (WAEF) which again achieves substantial gain over baseline. In WAEF1, WAEF2 and WAEF, we update x_m& y_m, y_m, and x_m, respectively. It is evident that WAEF performs better than its counterparts, which validates our claim of updating x_m alone. We report that the WAEF tracker exceeds the baseline with a gain of 1.24% in success rate, and 0.69% in precision.

In Figure 5.11, we compare our top-performing trackers with the state-of-the-art trackers. Among the compared trackers, our WAEF tracker does exceedingly well, outperforming the winner on OTB50. We observe that the proposed framework is robust enough to tackle the typical challenging issues in object tracking. In Table 5.18, we show the categorical comparison of area under the curve (AUC) and success rate, which are the standard metrics on benchmark results. The WAEF tracker provides substantial cumulative gain of 14.8% over all the crucial categories on OTB50. Moreover, the proposed architecture does not deteriorate the baseline performance in either of the aforementioned categories.

Tracker	WAEF	TREF	MDNet	ECO	CCOT	DeepSRDCF	SRDCF	HDT	KCF
Out of view	0.657	0.654	0.617	0.644	0.636	0.551	0.512	0.479	0.368
Occlusion	0.654	0.652	0.631	0.643	0.632	0.555	0.532	0.504	0.405
Illumination Variation	0.632	0.628	0.625	0.623	0.594	0.530	0.509	0.488	0.386
Low Resolution	0.626	0.623	0.608	0.617	0.613	0.511	0.486	0.471	0.334
Background Clutter	0.638	0.636	0.625	0.629	0.588	0.535	0.517	0.494	0.388
Deformation	0.634	0.634	0.627	0.621	0.602	0.532	0.520	0.488	0.399
Out-of-plane rotation	0.646	0.642	0.627	0.636	0.605	0.549	0.516	0.503	0.399
FastMotion	0.645	0.643	0.620	0.637	0.625	0.554	0.523	0.499	0.365

Table 5.7
The success and precision plots in various category of our proposed WAEF, TREF, and several state-of-the-art trackers on OTB50 dataset.

We also evaluate the WAEF tracker on VOT2016 dataset, and compare the results in Table 5.8. The WAEF tracker offers remarkable achievement, improving 5.28% AEO,

Tracker	WAEF	ECO_EF	MDNet	ECO	CCOT	DeepSRDCF	TricTRACK
AEO	0.3750	0.3616	0.3584	0.3563	0.3310	0.2763	0.1995
Ar	1.78	2.13	1.40	1.90	2.13	2.47	5.90
Rr	2.38	2.38	2.70	2.58	2.77	4.00	6.92

Table 5.8
Overall quantitative analysis of few trackers on VOT2016. AEO, Ar, and Rr represents average expected overlap, accuracy rank, and robustness rank, respectively.

6.31% accuracy rank, and 7.75% robustness rank relative to baseline. In particular, the WAEF tracker provides substantial improvement of 19.04% in occlusion, 27.66% in illumination change, 33.33% in empty, and 10% in size change category of VOT2016, as can be inferred from Figure 5.12. Also, to validate the usefulness of EF, we have experimented ECO with EF alone. We observe that the enhancement filter assists in shaping the visual information which eventually leads to notable gain in AEO. This implicates that the robust feature extractors still lack high quality visual inputs that may boost the performance. When all the contributions are incorporated, the GPU version of WAEF runs at 8 FPS, which is same as the baseline. Due to unavailability of results on whole OTB50 and VOT2016 in the required format, we were unable to compare a few similar regression models [34, 74, 85] with our WAEF tracker. However, since the WAEF tracker outperforms the state-of-the-art trackers, which are much better than the existing temporal regression models, we report that our proposed architecture will surpass these models with ease.

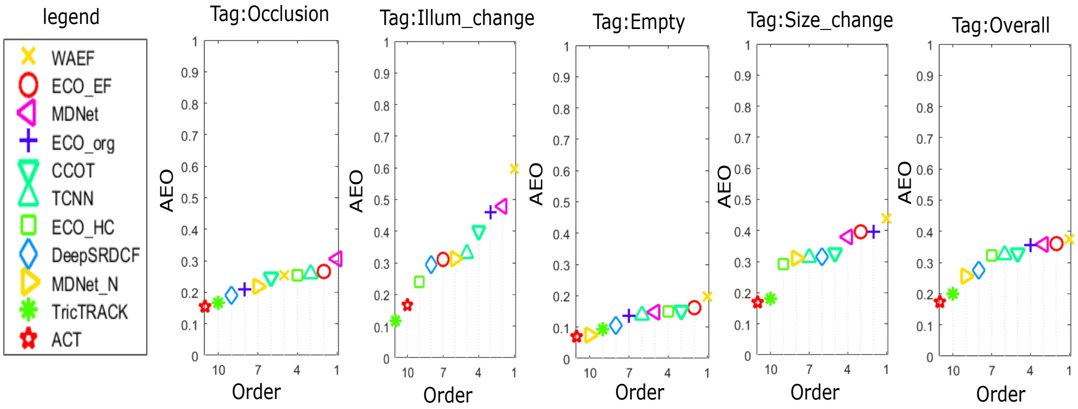

Figure 5.12: Average Expected Overlap (AEO) analysis of our WAEF tracker and several other state-of-the-art trackers in various challenging categories of VOT2016.

AR ranking, as can be inferred from Figure 5.13, captures the accuracy and robustness

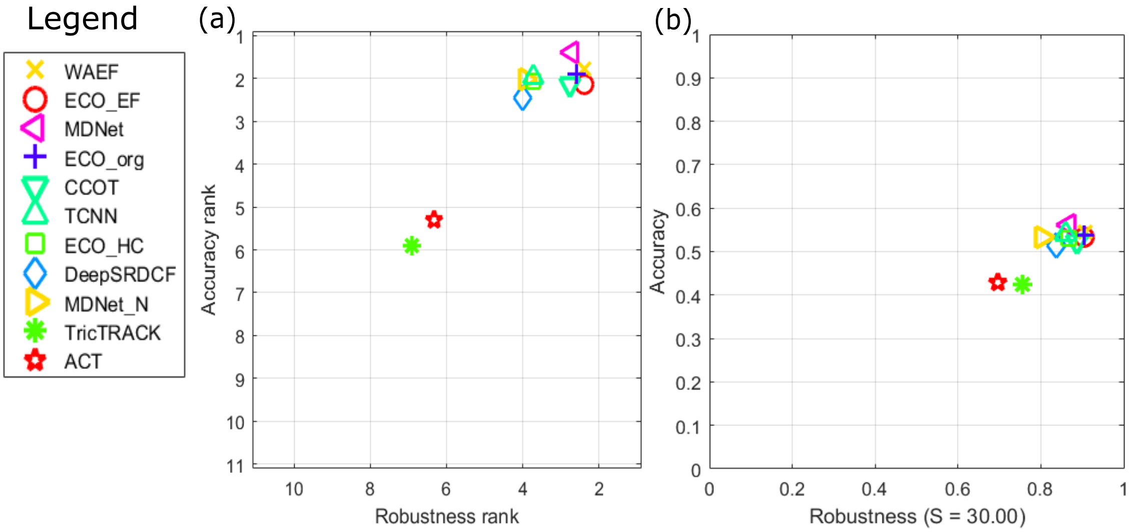

Figure 5.13: (a) Ranking plot for experiment baseline. (b) AR plot for experiment baseline. Highly accurate and highly robust trackers tend to fall at the top right corner. The proposed WAEF tracker (Ar = 1.78, Rr = 2.38, A = 0.55, R = 0.94) surpasses the baseline ECO (Ar = 1.90, Rr = 2.58, A = 0.54, R = 0.93) in accuracy as well as robustness which lead to considerable gain in overall performance.

of a given tracker over all the sequences from VOT benchmark dataset(here, VOT2016). Accuracy is a measure of overlap with ground truth bounding box whereas robustness is a measure of deviation from actual centroid.

5.4.1 Expected Overlap Analysis

Most of the existing trackers suffer from a common problem that arises when the number of frames in a sequence increases. The minor deviation from actual target gets accumulated over time which subsequently results in drifting into the background. Therefore, the expected overlap analysis plots, as shown in Figure 5.14, are used in determining the overall performance of these trackers in such scenarios. The proposed WAEF and ECO_EF trackers outperform the baseline ECO and several other state-of-the-art trackers in most of the categories on VOT2016 dataset. All the expected overlap curves are generated as such by the VOT toolkit, which does not offer the provision to measure area under these individual curves. Though there is a considerable improvement in almost all individual categories, the gain is significant in illumination change, size change, and empty categories, as can be inferred from Figure 5.15, Figure 5.16, and Figure 5.17, respectively.

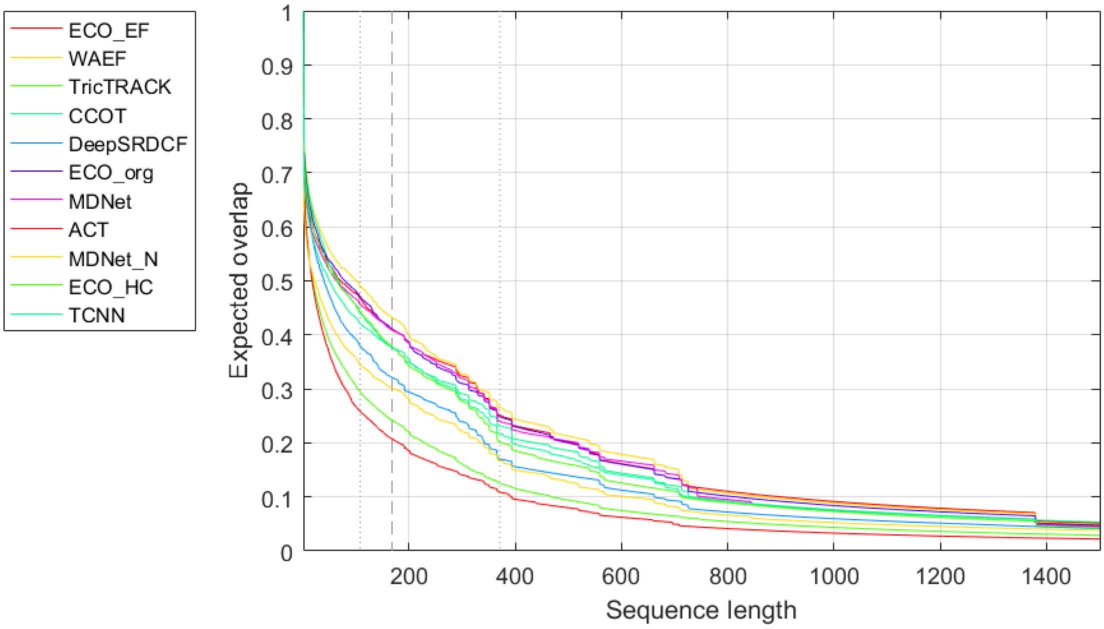

Figure 5.14: Expected overlap curves for baseline (Overall) on VOT2016 dataset.

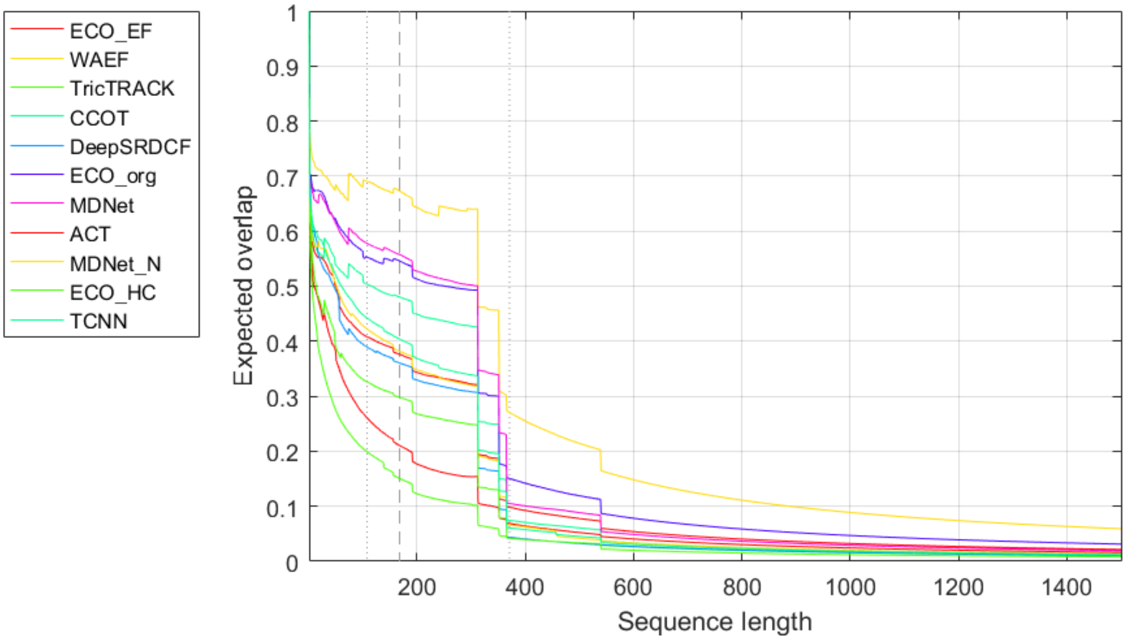

Figure 5.15: Expected overlap curves for baseline (illumination change) on VOT2016 dataset.

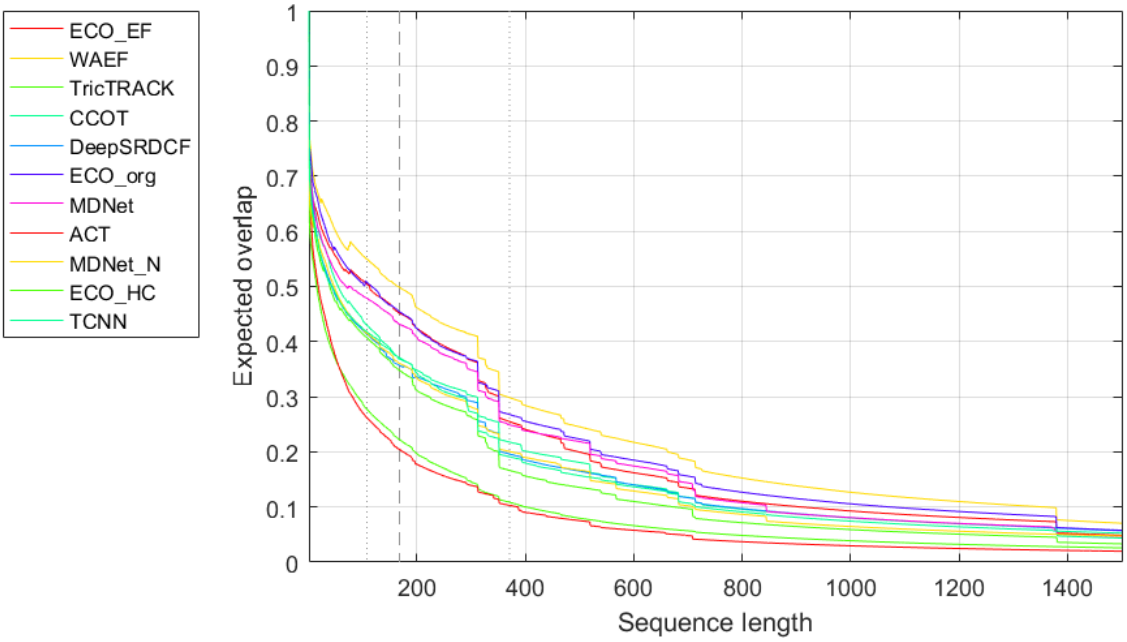

Figure 5.16: Expected overlap curves for baseline (size change) on VOT2016 dataset.

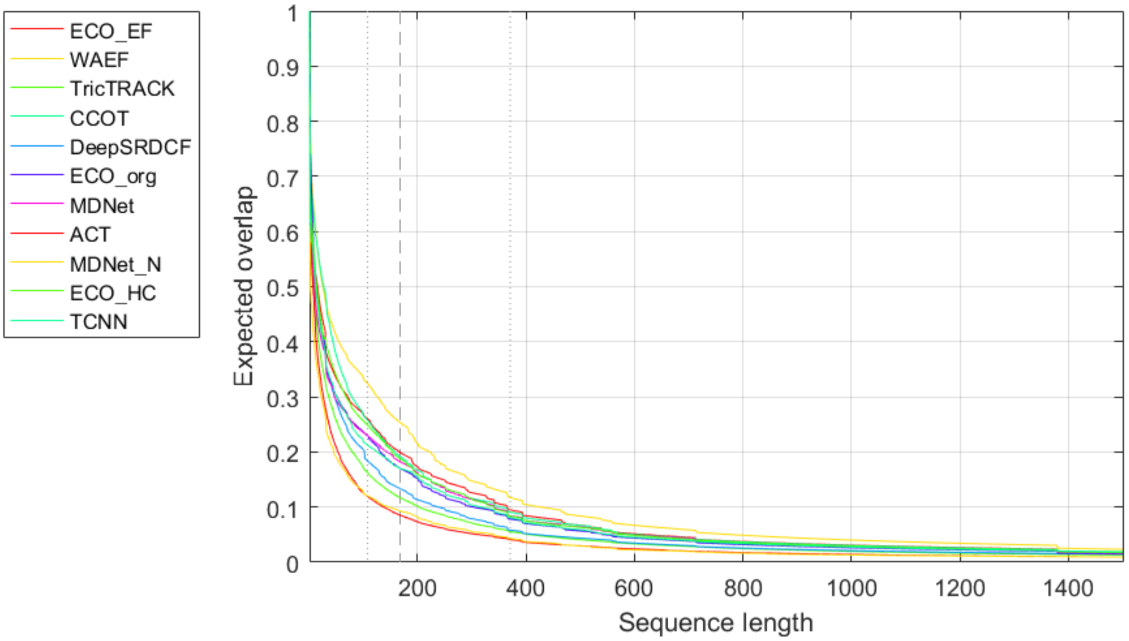

Figure 5.17: Expected overlap curves for baseline (empty) on VOT2016 dataset.

5.4.2 Quantitative Evaluation

The area under the curve (AUC) of success rate plot against overlap threshold is one of the major evaluation metrics to compare the performance of various trackers. The success plots of several challenging categories help to comprehend the overall performance of a tracker. From the Figure 5.18, it is evident that the overall performance of the proposed WAEF tracker is significantly better than the baseline ECO.

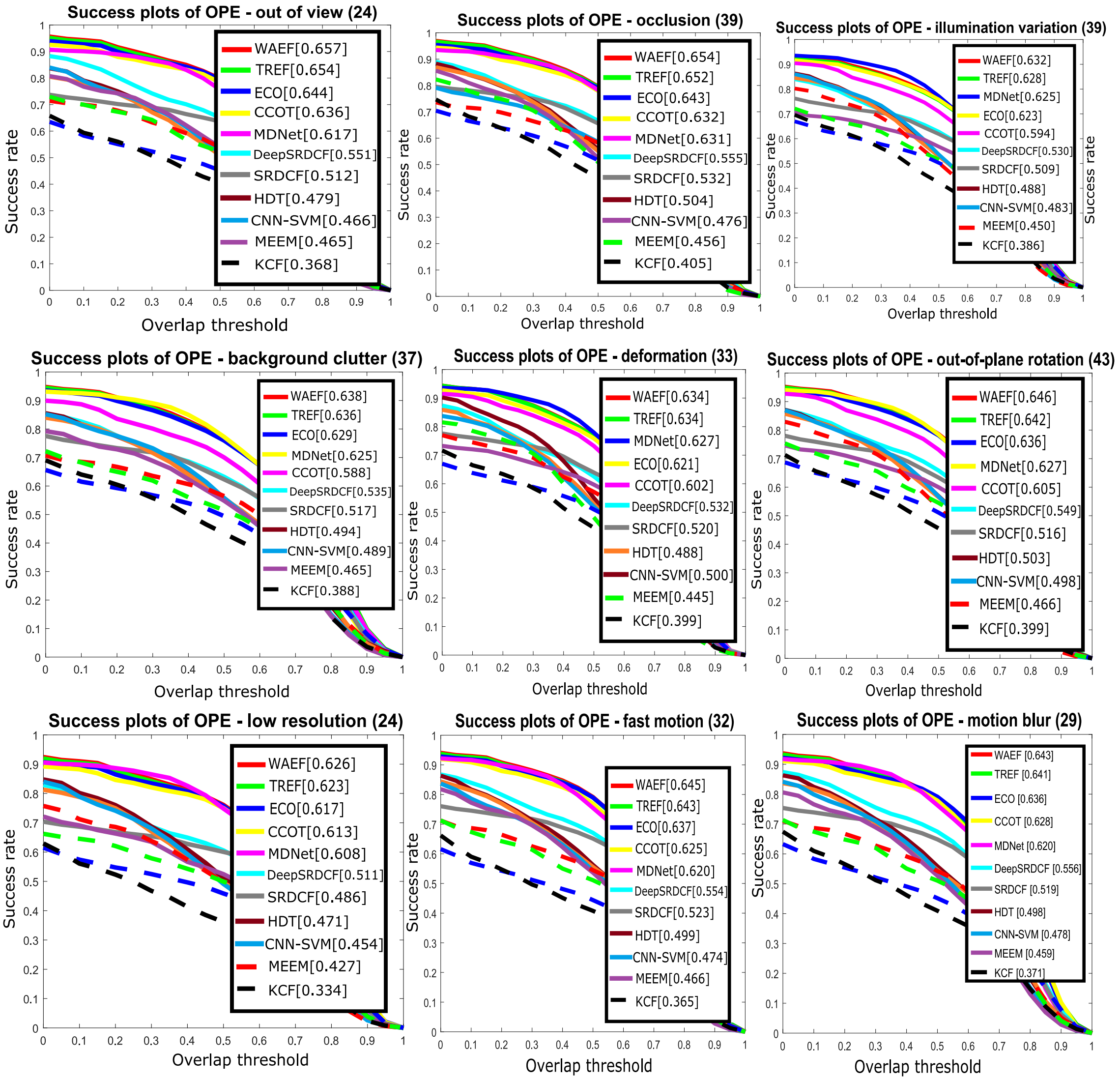

Figure 5.18: The success and precision plots of our proposed WAEF, TREF, and several state-of-the-art trackers on OTB50 dataset. The WAEF and TREF trackers perform favourably against the state-of-the-art trackers in all challenging scenarios.

5.4.3 Qualitative Analysis

To qualitatively assess the performance, we have compared the bounding boxes of various trackers including the proposed WAEF tracker in Figure 5.19. We have selected few frames from some of the challenging sequences from OTB50 where the categorical comparison can be easily assessable. The proposed WAEF tracker performs significantly better than the baseline approach (ECO) and other state-of-the-art trackers in several tough categories.

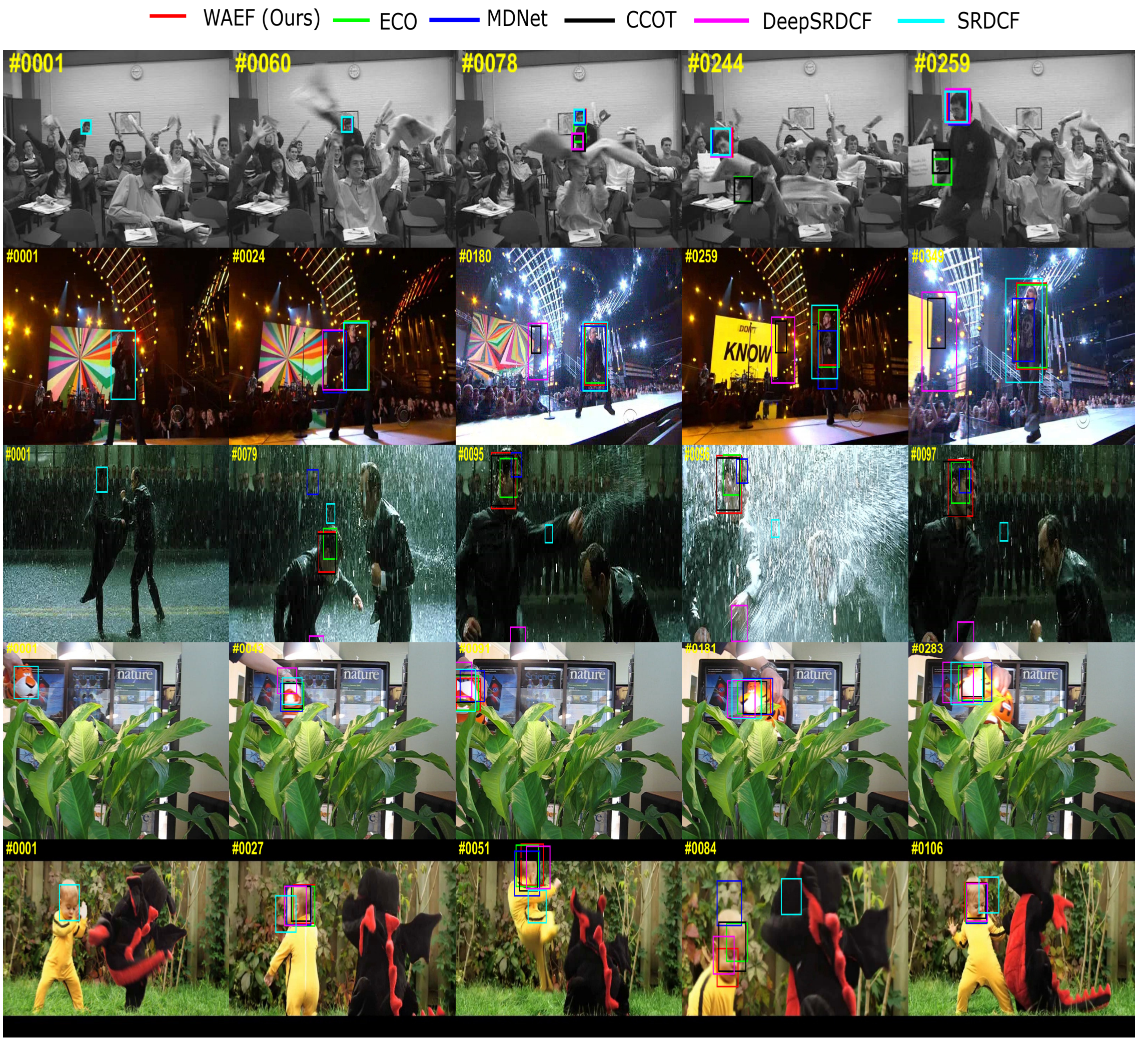

Figure 5.19: Qualitative comparison of the proposed WAEF tracker with other state-of-the-art trackers on few tough sequences from OTB50 dataset, such as (top to bottom) Freeman4, Singer2, Matrix, Tiger2, and Dragonbaby. The compared sequences validate the fact that WAEF tracker improves the performance of baseline ECO. It, in fact, provides substantial gain in various challenging categories with certain improvement in AUC of success rate as well as precision.

Chapter 6

Concluding Remarks and Future Scope

In our study of deep learning based trackers, we investigated the consequences of rotation adaptiveness in object tracking. The proposed consistency techniques surely outperformed the baseline deep learning based algorithms [9, 18]. The success rate improved by 4.6% whereas precision, by 6.75%, as given in Table 5.1. According to the evaluation of proposed Siamese DSR on VOT [5], a drastic improvement in robustness rank by 15.7% and accuracy rank by 14.3% was observed with no degradation in overlap ratio (Table 5.3). Above all, detecting the orientation of the target object, as proposed in this thesis, will certainly be a significant boost in the tracking paradigm. As per the analysis, the concept of rotation adaptive tracking with aforementioned motion consistencies has been exceptional in determining the target centroid in most of the tough sequences in popular tracking benchmarks. Our future research may include replacing the simple CNN present in both Siamese and CFnet architectures with a very deep CNN.

Thereafter, in our study of correlation filter based trackers, we demonstrated that employing a simple, yet effective image enhancement technique, prior to feature extraction, can yield considerable gain in visual object tracking. We analyzed the effectiveness of proposed rotation adaptive correlation filters in standard DCF formulation, and showed compelling results on a popular tracking benchmark. We renovated the sub-grid detection approach by incorporating false positive elimination, and object's orientation, which was reflected favourably in the overall performance. Also, the supervision of displacement consistency on CF trackers showed promising results in numerous challenging scenarios. Moreover, the proposed contributions are simple and straight forward, and can be suitably integrated with other CF trackers, leading to substantial improvement in overall performance. Our future research of rotation adaptive correlation filter trackers may include a more sophisticated optimization in the object's orientation space.

At the end, in our study of temporal regression based trackers, we analysed the impact of ridge regression with Tikhonov regularization in temporal domain, and showed promising results on popular benchmarks. Further, we introduced an approach to regress in the temporal domain based on weighted aggregation and entropy estimation, which could provide drastic improvement in various tracking benchmarks. Moreover, this temporal regression framework is generic, and can accommodate other detectors with simultaneously leveraging the spatial and temporal correspondence. Our future scope may include robust feature selection based on sophisticated density estimation. Also, we may integrate the proposed framework into other detectors, and analyse the impact on generic object tracking. Finally, we intend to write a research journal consolidating our whole constributions, which include the proposed deep learning based models, correlation filter based models, and temporal regression based models.

List of Papers based on this Thesis

- Rout, Litu, Sidhartha, Gorthi, RKSS Manyam, and Mishra, Deepak. "Rotation Adaptive Visual Object Tracking with Motion Consistency", in 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1047-1055, March, 2018. *[Status: Published]*

- Rout, Litu, Mishra, Deepak , and Gorthi, RKSS Manyam. "WAEF: Weighted Aggregation with Enhancement Filter for Visual Object Tracking", in European Conference on Computer Vision (ECCV), September, 2018. *[Status: Under Peer Review]*

- Rout, Litu, Raju, Priya Mariam, Mishra, Deepak, and Gorthi, RKSS Manyam. "Rotation Adaptive and Illumination Invariant Correlation Filter Tracker with False Positive Elimination", in British Machine Vision Conference (BMVC), September, 2018. *[Status: Under Peer Review]*

- Rout, Litu, Raju, Priya Mariam, Mishra, Deepak, and Gorthi, RKSS Manyam. "Boosting Visual Object Tracking performance using a stack of Signal Processing and Machine Learning algorithms", in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2018. *[Status: To be submitted]*

- Rout, Litu, Mishra, Deepak, and Gorthi, RKSS Manyam. "Visual Object Tracking workshop", in European Conference on Computer Vision (ECCV), 2018. *[status: To be submitted]*

Bibliography

Bibliography

[1] Marr, David. 1982. Vision: a computational investigation into the human representation and processing of visual information. San Francisco: W.H. Freeman.

[2] Blake, Andrew, and Alan Yuille. "Active Vision. Artificial Intelligence." (1992).

[3] Aloimonos, John, Isaac Weiss, and Amit Bandyopadhyay. "Active vision." International journal of computer vision 1, no. 4 (1988): 333-356.

[4] Tao, Ran, Efstratios Gavves, and Arnold WM Smeulders. “Siamese instance search for tracking”, Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on. IEEE, 2016.

[5] Kristan, Matej, Jiri Matas, Aleš Leonardis, Tomáš Vojíř, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. "A novel performance evaluation methodology for single-target trackers." IEEE transactions on pattern analysis and machine intelligence 38, no. 11 (2016): 2137-2155.

[6] Henriques, João F., Rui Caseiro, Pedro Martins, and Jorge Batista. "High-speed tracking with kernelized correlation filters." IEEE Transactions on Pattern Analysis and Machine Intelligence 37, no. 3 (2015): 583-596.

[7] Danelljan, Martin, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. "Learning spatially regularized correlation filters for visual tracking." In Proceedings of the IEEE International Conference on Computer Vision, pp. 4310-4318. 2015.

[8] Danelljan, Martin, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. "Beyond correlation filters: Learning continuous convolution operators for visual tracking." In European Conference on Computer Vision, pp. 472-488. Springer, Cham, 2016.

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

[10] Danelljan, Martin, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. "Convolutional features for correlation filter based visual tracking." In Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 58-66. 2015.

[11] Nam, Hyeonseob, Mooyeol Baek, and Bohyung Han. "Modeling and propagating cnns in a tree structure for visual tracking." arXiv preprint arXiv:1608.07242 (2016).

[12] Nam, Hyeonseob, and Bohyung Han. "Learning multi-domain convolutional neural networks for visual tracking." In Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, pp. 4293-4302. IEEE, 2016.

[13] Wang, Lijun, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. "Visual tracking with fully convolutional networks." In Proceedings of the IEEE International Conference on Computer Vision, pp. 3119-3127. 2015.

[14] Wu, Yi, Jongwoo Lim, and Ming-Hsuan Yang. "Object tracking benchmark." IEEE Transactions on Pattern Analysis and Machine Intelligence 37, no. 9 (2015): 1834-1848.

[15] Bertinetto, Luca, João F. Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. "Learning feed-forward one-shot learners." In Advances in Neural Information Processing Systems, pp. 523-531. 2016.

[16] Zagoruyko, Sergey, and Nikos Komodakis. "Learning to compare image patches via convolutional neural networks." In Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, pp. 4353-4361. IEEE, 2015.

[17] Li, Yang, and Jianke Zhu. "A scale adaptive kernel correlation filter tracker with feature integration." In European Conference on Computer Vision, pp. 254-265. Springer, Cham, 2014.

[18] Valmadre, Jack, Luca Bertinetto, João Henriques, Andrea Vedaldi, and Philip HS Torr. "End-to-end representation learning for correlation filter based tracking." In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pp. 5000-5008. IEEE, 2017.

[19] Adam, Amit, Ehud Rivlin, and Ilan Shimshoni. “Robust fragments-based tracking using the integral histogram”, In Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on, vol. 1, pp. 798-805. IEEE, 2006.

[20] Comaniciu, Dorin, Visvanathan Ramesh, and Peter Meer. “Kernel-based object tracking”, IEEE Transactions on pattern analysis and machine intelligence 25, no. 5 (2003): 564-577.

[21] Kwon, Junseok, and Kyoung Mu Lee. “Visual tracking decomposition”, In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 1269-1276. IEEE, 2010.

[22] Zhang, Tianzhu, Si Liu, Narendra Ahuja, Ming-Hsuan Yang, and Bernard Ghanem. “Robust visual tracking via consistent low-rank sparse learning”, International Journal of Computer Vision 111, no. 2 (2015): 171-190.

[23] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. “Tracking-learning-detection”, IEEE transactions on pattern analysis and machine intelligence 34, no. 7 (2012): 1409-1422.

[24] Wang, Shu, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang. “Superpixel tracking”, In Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 1323-1330. IEEE, 2011.

[25] Babenko, Boris, Ming-Hsuan Yang, and Serge Belongie. “Robust object tracking with online multiple instance learning”, IEEE transactions on pattern analysis and machine intelligence 33, no. 8 (2011): 1619-1632.

[26] Henriques, João F., Rui Caseiro, Pedro Martins, and Jorge Batista. "High-Speed Tracking with Kernelized Correlation Filters."

[27] Liu, Si, Tianzhu Zhang, Xiaochun Cao and Changsheng Xu. “Structural correlation filter for robust visual tracking”, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4312-4320. 2016.

[28] Danelljan, Martin, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. “Learning spatially regularized correlation filters for visual tracking”, In Proceedings of the IEEE International Conference on Computer Vision, pp. 4310-4318. 2015.

[29] Mei, Xue, and Haibin Ling. “Robust visual tracking and vehicle classification via sparse representation”, IEEE transactions on pattern analysis and machine intelligence 33.11 (2011): 2259-2272.

[30] Bengio, Yoshua, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives”, IEEE transactions on pattern analysis and machine intelligence 35.8 (2013): 1798-1828.

[31] Pérez, Patrick, Carine Hue, Jaco Vermaak, and Michel Gangnet. "Color-based probabilistic tracking." In European Conference on Computer Vision, pp. 661-675. Springer, Berlin, Heidelberg, 2002.

[32] Zhou, Huiyu, Yuan Yuan, and Chunmei Shi. “Object tracking using SIFT features and mean shift”, Computer vision and image understanding 113.3 (2009): 345-352.

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

[34] Liu, Ting, Gang Wang, and Qingxiong Yang. “Real-time part-based visual tracking via adaptive correlation filters”, Intelligence 2345390 (2015).

[35] Rout, Litu, Gorthi RKSS Manyam, and Deepak Mishra. “Rotation Adaptive Visual Object Tracking with Motion Consistency”, arXiv preprint arXiv:1709.06057 (2017).

[36] Kwon, Junseok, and Kyoung Mu Lee. “Tracking by sampling trackers”, Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

[37] Danelljan, Martin, Fahad Shahbaz Khan, Michael Felsberg, and Joost Van de Weijer. "Adaptive color attributes for real-time visual tracking." In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, Ohio, USA, June 24-27, 2014, pp. 1090-1097. IEEE Computer Society, 2014.

[38] Danelljan, Martin, Goutam Bhat, F. Shahbaz Khan, and Michael Felsberg. "ECO: efficient convolution operators for tracking." In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 21-26. 2017.

[39] Yilmaz, Alper, Omar Javed, and Mubarak Shah. "Object tracking: A survey." Acm computing surveys (CSUR) 38, no. 4 (2006): 13.

[40] Aggarwal, Jake K., and Quin Cai. "Human motion analysis: A review." Computer vision and image understanding 73, no. 3 (1999): 428-440.

[41] Gavrila, Dariu M. "The visual analysis of human movement: A survey." Computer vision and image understanding 73, no. 1 (1999): 82-98.

[42] Moeslund, Thomas B., and Erik Granum. "A survey of computer vision-based human motion capture." Computer vision and image understanding 81, no. 3 (2001): 231-268.

[43] Comaniciu, Dorin, Visvanathan Ramesh, and Peter Meer. “Kernel-based object tracking.” IEEE Transactions on pattern analysis and machine intelligence 25.5 (2003): 564-577.

[44] Lee, Kuang-Chih, Jeffrey Ho, Ming-Hsuan Yang, and David Kriegman. "Visual tracking and recognition using probabilistic appearance manifolds." Computer Vision and Image Understanding 99, no. 3 (2005): 303-331.

[45] Mei, Xue, and Haibin Ling. “Robust visual tracking and vehicle classification via sparse representation.” IEEE transactions on pattern analysis and machine intelligence 33.11 (2011): 2259-2272.

[46] Collins, Robert T., Yanxi Liu, and Marius Leordeanu. “Online selection of discriminative tracking features.” IEEE transactions on pattern analysis and machine intelligence 27.10 (2005): 1631-1643.

[47] Birchfield, Stan. “Elliptical head tracking using intensity gradients and color histograms.” Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on. IEEE, 1998.

[48] Badrinarayanan, Vijay, Patrick Perez, Francois Le Clerc, and Lionel Oisel. "Probabilistic color and adaptive multi-feature tracking with dynamically switched priority between cues." In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pp. 1-8. IEEE, 2007.

[49] Park, Dong Woo, Junseok Kwon, and Kyoung Mu Lee. “Robust visual tracking using autoregressive hidden Markov model.” Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

[50] Stenger, Bjorn, Thomas Woodley, and Roberto Cipolla. "Learning to track with multiple observers." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

[51] Stenger, Bjorn, Thomas Woodley, and Roberto Cipolla. "Learning to track with multiple observers." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

[52] Koch, Gregory, Richard Zemel, and Ruslan Salakhutdinov. "Siamese neural networks for one-shot image recognition." ICML Deep Learning Workshop. Vol. 2. 2015.

[53] Hua, Yang, Karteek Alahari, and Cordelia Schmid. "Online object tracking with proposal selection." In Computer Vision (ICCV), 2015 IEEE International Conference on, pp. 3092-3100. IEEE, 2015.

[54] Rowley, Henry A., Shumeet Baluja, and Takeo Kanade. "Rotation invariant neural network-based face detection." Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on. IEEE, 1998.

[55] Burel, Gilles, and Dominique Carel. "Detection and localization of faces on digital images." Pattern Recognition Letters 15.10 (1994): 963-967.

[56] Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems. 2015.

[57] Wah, Catherine, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. "The caltech-ucsd birds-200-2011 dataset." (2011).

[58] Bolme, David S., J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui. "Visual object tracking using adaptive correlation filters." In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 2544-2550. IEEE, 2010.

[59] Nummiaro, Katja, Esther Koller-Meier, and Luc Van Gool. "An adaptive color-based particle filter." Image and vision computing 21, no. 1 (2003): 99-110.

[60] Oron, Shaul, Aharon Bar-Hillel, Dan Levi, and Shai Avidan. "Locally orderless tracking." International Journal of Computer Vision 111, no. 2 (2015): 213-228.

[61] Ma, Chao, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. "Hierarchical convolutional features for visual tracking." In Proceedings of the IEEE International Conference on Computer Vision, pp. 3074-3082. 2015.

[62] Zhang, Mengdan, Junliang Xing, Jin Gao, Xinchu Shi, Qiang Wang, and Weiming Hu. "Joint scale-spatial correlation tracking with adaptive rotation estimation." In Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 32-40. 2015.

[63] Dong, Yanmei, Min Yang, and Mingtao Pei. "Visual tracking with sparse correlation filters." In Image Processing (ICIP), 2016 IEEE International Conference on, pp. 439-443. IEEE, 2016.

[64] Sun, Chong, Huchuan Lu, and Ming-Hsuan Yang. "Learning Spatial-Aware Regressions for Visual Tracking." arXiv preprint arXiv:1706.07457 (2017).

[65] He, Zhiqun, Yingruo Fan, Junfei Zhuang, Yuan Dong, and HongLiang Bai. "Correlation Filters With Weighted Convolution Responses." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1992-2000. 2017.

[66] Lukezic, Alan, Tomás Vojír, L. Cehovin Zajc, Jiri Matas, and Matej Kristan. "Discriminative correlation filter with channel and spatial reliability." In IEEE Conf. on Computer Vision and Pattern Recognition, pp. 4847-4856. 2017.

[67] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 886-893. IEEE, 2005.

[68] Wen, Longyin, Zhaowei Cai, Zhen Lei, Dong Yi, and Stan Z. Li. "Robust online learned spatio-temporal context model for visual tracking." IEEE Transactions on Image Processing 23, no. 2 (2014): 785-796.

[69] Cui, Zhen, Shengtao Xiao, Jiashi Feng, and Shuicheng Yan. "Recurrently target-attending tracking." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1449-1458. 2016.

[70] Teng, Zhu, Junliang Xing, Qiang Wang, Congyan Lang, Songhe Feng, and Yi Jin. "Robust object tracking based on temporal and spatial deep networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1144-1153. 2017.

[71] Dinh, Thang Ba, Nam Vo, and Gérard Medioni. "Context tracker: Exploring supporters and distracters in unconstrained environments." In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 1177-1184. IEEE, 2011.

[72] Wang, Xiaomeng, Michel Valstar, Brais Martinez, Muhammad Haris Khan, and Tony Pridmore. "Tric-track: Tracking by regression with incrementally learned cascades." In Proceedings of the IEEE International Conference on Computer Vision, pp. 4337-4345. 2015.

[73] Martinez, Brais, Michel F. Valstar, Xavier Binefa, and Maja Pantic. "Local evidence aggregation for regression-based facial point detection." IEEE transactions on pattern analysis and machine intelligence 35, no. 5 (2013): 1149-1163.

[74] Ning, Guanghan, Zhi Zhang, Chen Huang, Xiaobo Ren, Haohong Wang, Canhui Cai, and Zhihai He. "Spatially supervised recurrent convolutional neural networks for visual object tracking." In Circuits and Systems (ISCAS), 2017 IEEE International Symposium on, pp. 1-4. IEEE, 2017.

[75] Petrou, Maria, and Costas Petrou. Image processing: the fundamentals. John Wiley & Sons, 2010.

[76] Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. Deep learning. Vol. 1. Cambridge: MIT press, 2016.

[77] Li, Fei-Fei, Rob Fergus, and Pietro Perona. "One-shot learning of object categories." IEEE Transactions on Pattern Analysis and Machine Intelligence 28.4 (2006): 594-611.

[78] Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes (voc) challenge." International journal of computer vision 88, no. 2 (2010): 303-338.

[79] Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang et al. "Imagenet large scale visual recognition challenge." International Journal of Computer Vision 115, no. 3 (2015): 211-252.

[80] Smeulders, Arnold WM, Dung M. Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. "Visual tracking: An experimental survey." IEEE transactions on pattern analysis and machine intelligence 36, no. 7 (2014): 1442-1468.

[81] Chu, Dung M., and Arnold WM Smeulders. "Thirteen hard cases in visual tracking." Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on. IEEE, 2010.

[82] Song, Shuran, and Jianxiong Xiao. "Tracking revisited using RGBD camera: Unified benchmark and baselines." Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, 2013.

[83] Shotton, Jamie, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. "Real-time human pose recognition in parts from single depth images." In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 1297-1304. Ieee, 2011.

[84] Cehovin, Luka, Matej Kristan, and Ales Leonardis. "Is my new tracker really better than yours?." Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on. IEEE, 2014.

[85] Chen, Kai, and Wenbing Tao. "Convolutional regression for visual tracking." IEEE Transactions on Image Processing (2018).