Generic Approaches to Enhanced Correlation Filter Tracking

A Thesis submitted

in partial fulfilment for the degree of

Doctor of Philosophy

by

Priya Mariam Raju

Department of Avionics

INDIAN INSTITUTE OF SPACE SCIENCE AND TECHNOLOGY

THIRUVANANTHAPURAM - 695 547

March 2023

CERTIFICATE

This is to certify that the thesis titled ***Generic Approaches to Enhanced Correlation Filter Tracking*** submitted by **Priya Mariam Raju**, to the Indian Institute of Space Science and Technology, Thiruvananthapuram, in partial fulfilment for the award of the degree of **Doctor of Philosophy**, is a bonafide record of the original work carried out by her under my supervision. The contents of this thesis, 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
Research Supervisor
Professor
Department of Avionics
IIST

Dr. N. Selvaganesan
Professor and Head
Department of Avionics
IIST

Place: IIST, Thiruvananthapuram
Date: March 31, 2023

Declaration

I declare that this thesis titled ***Generic Approaches to Enhanced Correlation Filter Tracking*** submitted in partial fulfilment for the award of the degree of **Doctor of Philosophy** is a record of the original work carried out by me under the supervision of **Dr. Deepak Mishra** and has not formed the basis for the award of any degree, diploma, associateship, 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 acknowledgments have been made wherever the findings of others have been cited.

Place: IIST, Thiruvananthapuram
Date: March 31, 2023

Priya Mariam Raju
SC16D051

Acknowledgements

First, I would like to thank God Almighty for blessing me with the opportunity, knowledge and health to accomplish the thesis.

I would like to express my sincere gratitude to my research supervisor, Dr. Deepak Mishra, for his valuable guidance, dedication, time, and constant support throughout my research study. I will be grateful forever that he took me on as a student and continued to have faith in me over the years.

I would also like to thank Dr. Rama Krishna Sai Subrahmanyam Gorthi (Associate Professor, Department of Electrical Engineering, IIT Tirupati), whose expertise and suggestions helped me to start my research journey in object tracking. I also whole-heartily thank him for the continuous support and advice extended to me at all needed times.

I also express my sincere gratitude to Dr. Prerana Mukherjee (Assistant Professor, School of Engineering, Jawaharlal Nehru University, New Delhi) for guiding and supporting me in the research progress. I thankfully remember all the efforts she has taken to help me, particularly during the lockdown time.

I would also like to thank the members of my doctoral committee: Dr. B.S. Manoj (DC chairperson, Professor, Department of Avionics, IIST), Dr. R. Venkatesh Babu (Professor, Department of Computational and Data Sciences, IISc, Bangalore), Dr. Kurien Issac K. (Senior Professor, Department of Aerospace, IIST) and Dr. Sheeba Rani J. (Associate Professor, Department of Avionics, IIST), who were always supportive at different stages of my research and helped me progress with their valuable suggestions and constructive criticism.

I also express my gratitude and thanks to Director, IIST, Dean Academics, and Dean R and D for providing me the research ambience during my tenure in IIST.

I thankfully acknowledge my friend and fellow researcher Ms. Aswathy P. for all the discussions and support throughout my research and for her help and enthusiasm in developing a good research ambience in the Virtual Reality (VR) lab.

I express my thanks and wishes to my friends and colleagues Ms. Soumya S., Ms. Deepthy Sivan, Ms. Elizabeth George, Ms. Jinu Joseph, Mr. G. Gopakumar, Mr. Rahul Waghmare, Mr. Vinod Kumar P., Ms. Neethu S., Ms. Pallavi Venugopal, Ms. Aayushi Jain, Ms. Swetha V.C., Ms. Manne Ruchitha, Ms. Minha Mubarak, Ms. Soumya Sara John, Ms. Preethisree, Ms. Rejitha, Ms. Teja and all other friends of the VR lab fraternity for all their love, help and support that has made my work and life at IIST a wonderful time. The environment was often filled with academic discussions and the interactions with them will always be cherished. I also thank my other friends, technical and non-technical staff of various departments for giving a good association to me during the college days.

Last but not the least, I thank my parents, my husband, my younger sister and my in-laws for their constant prayers, support and encouragement that kept me motivated through various phases of my research.

Abstract

Automated video analysis and object tracking algorithms have attracted much attention in the recent years. Among the numerous tracking algorithms available, correlation-filter (CF) based discriminative trackers have become quite popular due to their simplicity and efficiency. Even though several tracking algorithms based on correlation filters are available today, most of them need techniques to successfully track an object whose appearance is constantly changing. Despite outstanding real-time tracking performance, correlation filter-based object tracking systems have several drawbacks. Some major challenges of correlation filter-based single object trackers are the object deformations brought on by rotation and scale changes, model drift problems, long-term tracking capabilities, and tracking resumption. The proposed research attempts to provide "generic approaches towards enhancing correlation filter trackers" considering these difficulties.

The initial stage of the thesis identifies the necessity of oriented bounding boxes to represent the target object. Changes in object orientation can result in significant changes in the object appearance relative to its initial appearance. These changes limit the solid training features collected from each video frame. Knowledge of object orientation can adapt the tracker to rotations and deformations. To effectively learn the target appearance model, the proposed research focuses on determining the orientation of the target object and generating oriented bounding boxes. The cost function of the correlation filter tracker is optimized across oriented samples to prevent false positives. Additionally, a localization method that considers the target displacement is used. The effectiveness of the proposed rotation adaptive correlation filter is analysed and has obtained promising results on popular tracking benchmarks.

The proposed research further examines the requirement for a re-initialization technique to handle tracking resumption and long-term tracking capabilities in CF trackers. To address this issue, an adaptive threshold and an online learning detector are developed which detects and overcomes tracking failures. The correlation tracker framework is incorporated with a detector-based re-initialization, significantly reducing model drift, and enhancing CF tracker's ability for long-term tracking. Through extensive evaluations, both qualitative

and quantitative, the proposed re-initialization scheme was observed to outperform the state-of-the-art trackers in accuracy and robustness under most of the challenging conditions.

Subsequently, the conventional recursive search technique in the correlation trackers was identified in the proposed research as a significant contributor to model drift. To minimize this problem, a segmentation-guided attention that creates highly reliable search regions, is integrated into the object tracking framework. These search regions are segmentation masks and are employed as a guiding step in tracking due to their superior localization capabilities. Target-level information is added to the tracking pipeline through a domain adaptation technique corresponding to each video sequence.

Finally, considering the superiority of the human visual attention in continually tracking an object and discriminating it from similar nearby objects, the proposed research establishes the significance of visual attention in object tracking. The subsequent proposal integrates correlation and visual attention on neighbouring frames to localize the target object across a video sequence and provide object representations in the form of bounding boxes and object segmentations.

The proposed tracking algorithms are evaluated on the popular benchmark datasets and have obtained competing performance with the state-of-the-art tracking algorithms.

Table of contents

List of figures

List of tables

Abbreviations

ACFN	Attentional Correlation Filter Network
AO	Average Overlap
AR	Augmented Reality
ARC	Aspect Ratio Changes
AUC	Area Under Curve
BACF	Background Aware Correlation Filter
BC	Background Clutter
BoVW	Bag-of-Visual-Words
CAM-DOT	Co-Attention Maps for Discriminative Object Tracking
CF	Correlation Filter
CFT	Correlation Filter Tracker
CLE	Center Location Error
CM	Camera Motion
CN	ColorNames
CNN	Convolutional Neural Network
CSK	Circulant Structure with Kernel
DA–SACOT	Domain Adaptive - SACOT
DCF	Discriminative Correlation Filter
DEF	Deformation
DFT	Discrete Fourier Transform
DLT	Detection-based Long-term Tracking
DMSRDCF	Deep Motion SRDCF
DSST	Discriminative Scale Space Tracking
EAO	Expected Average Overlap
ECO	Efficient Convolutional Operators
FFT	Fast Fourier Transform
FM	Fast Motion
FPE	False Positive Elimination

GPU	Graphical Processing Unit
HCI	Human Computer Interaction
HDT	Hedged Deep Tracking
HoG	Histogram of Gradients
IFFT	Inverse Fast Fourier Transform
IC	Illumination Correction
IoU	Intersection over Union
IPR	In Plane Rotation
IV	Illumination Variation
ITS	Intelligent Transportation System
KCF	Kernelized Correlation Filters
LCT	Long-term Correlation Tracking
LR	Low Resolution
MB	Motion Blur
MCPF	Multi-task Correlation Particle Filter
MRCNN	Mask RCNN
NCFT	Non-Correlation Filter Tracker
OCC	Occlusion
OPR	Out of Plane Rotation
OPE	One-Pass Evaluation
OTB	Object Tracking Benchmark
OV	Out of View
PO	Partial Occlusion
R-CFTs	Regularized Correlation Filter Trackers
RCNN	Regions with Convolutional Neural Networks
RPM	Region Proposal Module
SACOT	Segmentation guided Attention for Correlation based Object tracking
SDR	Score to Distance Ratio
SOT	Single Object Tracking
SRDCF	Spatially Regularized DCF
SRE	Spatial Robustness Evaluation
SV	Scale Variations
TC	TempleColor
TLM	Target Localization Module
TPU	Tensor Processing Unit
TRE	Temporal Robustness Evaluation

UAV	Unmanned Aerial Vehicle
VC	Viewpoint Changes
VOT	Visual Object Tracking

Chapter 1

Introduction

1.1 Visual Object Tracking (VOT)

Single object tracking (SOT), popularly known as Visual Object Tracking (VOT) has emerged as an essential topic in computer vision due to its applications in surveillance and security systems [4], traffic monitoring [5], biomedical systems [6], activity recognition [7], autonomous driving systems [8], human-computer interaction [9] and many other fields. Given the initial state of a target (target center and target size) in the first frame of a video sequence, object tracking aims to automatically obtain the object's state in the subsequent video frames. A visual indicator, such as an enclosing rectangle, surrounds the target indicating the object's location in a frame. A single object tracker is given the bounding box of the target in the first frame, and hence SOT falls under the category of detection-free tracking. The goal of the tracker is to determine the same target in all the remaining frames. Single object trackers should track any given object class even if no pre-trained model for the new class of object is available. The target state and its surrounding environment continually change in a video sequence, making it difficult to extract features, build models, and require more robust and accurate trackers.

Imposing limitations on objects' mobility and appearance can make tracking easier. Almost all tracking methods, for example, assume that the object's motion is smooth and without abrupt shifts. Based on apriori information, the object motion can be further constrained to have a constant velocity or constant acceleration. Prior knowledge of objects' size, shape, and appearance can help simplify the challenge.

Target initialization, appearance modeling, motion prediction, and target localization are the four sequential components of VOT. Target initialization annotates an object with representations like the centroid, a bounding box, an ellipse, contours etc. Appearance modeling entails detecting features for better region representation and efficiently creating

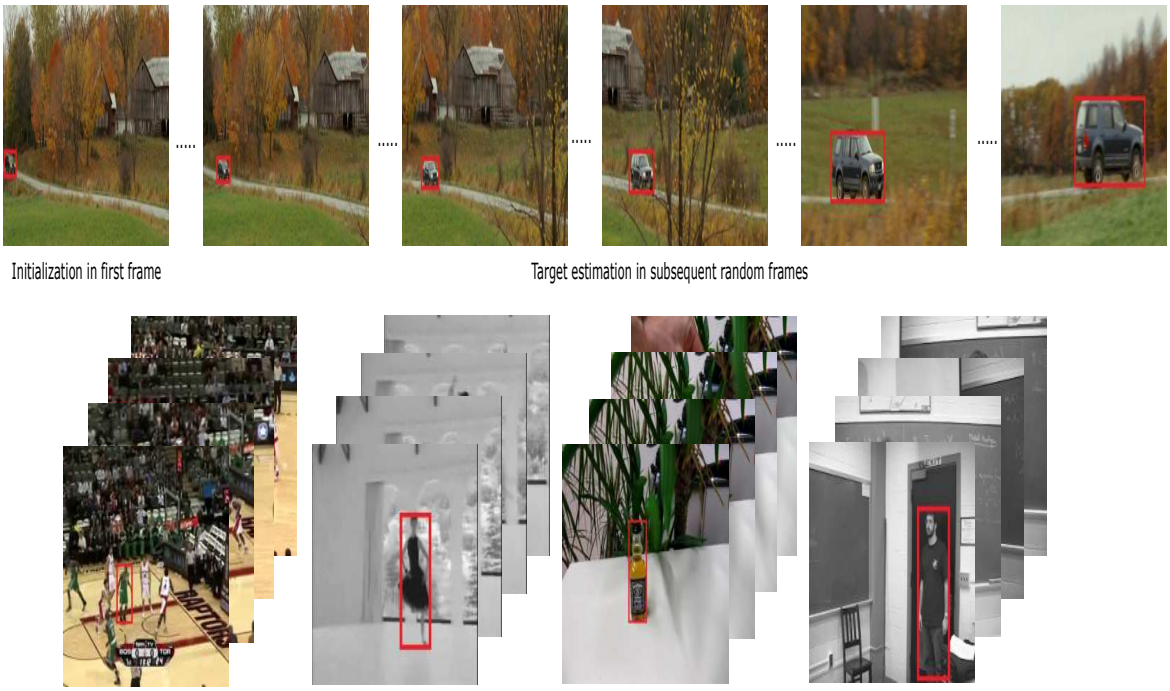


Fig. 1.1 Illustration of Single Object Tracking on sample sequences

mathematical models for object recognition. Motion prediction localizes the target in all the proceeding frames. Maximum posterior estimation, or greedy search, is used in the target positioning operation. The appearance and motion models can be constrained to solve tracking problems. These models are updated during tracking to incorporate new target appearances. Figure 1.1 illustrates the generic procedure of single object tracking on sample video sequences.

1.1.1 Object Representations

A target object is anything that is of relevance for subsequent analysis in the tracking scenario. For example, vehicles on the road, players in a sports scene, birds flying in the sky, people on the street are objects that may be crucial to tracking in a particular area. Object representations use the shape and appearance of the objects. This section discusses the typical object shape representations used in the benchmark datasets and evaluation toolkits, as shown in Figure 1.2.

1. **Points:** A single point, usually the object's centroid, or a collection of points represent the object. The point representation works well for tracking objects in small areas of an image.

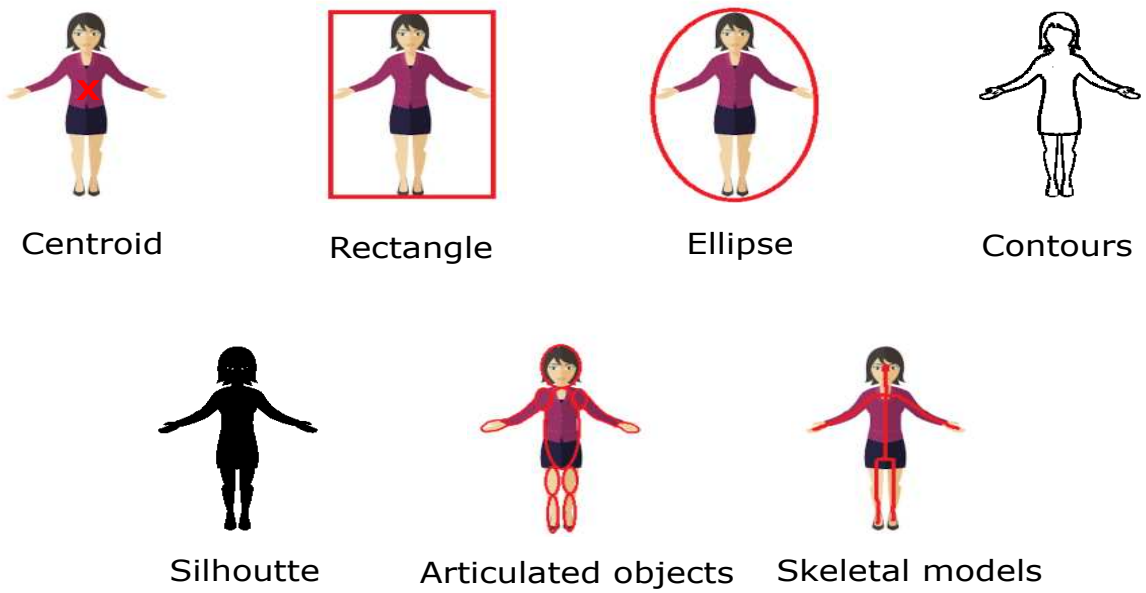


Fig. 1.2 Different target representations in the object tracking scenario.

2. **Geometric shapes:** Rectangles, ellipses, and other shapes also represent objects - translations, affine, or projective transformations model the object motion in such representations. The primitive geometric shapes can track both rigid and non-rigid objects.

3. **Object Contours and Silhouette:** The boundary of an object is its contour representation. The object's silhouette refers to the area inside the contour. Complex non-rigid shapes use silhouette and contour representations.

4. **Articulated shape models:** Articulated objects have different components connected by joints. The human body, for example, is an articulated object with joints connecting the different body parts. The constituent pieces of an articulated object use cylinders or ellipses for representation.

5. **Skeletal models:** The object skeleton is a transformation of the object silhouette along the medial axis. Object recognition primarily uses skeletal models for shape representation. Both articulated and rigid objects use skeletal representation.

1.1.2 Applications of Object Tracking

Object tracking is now possible with a wide range of computer vision applications. In recent years, imaging technology has advanced significantly. Cameras are now smaller, less

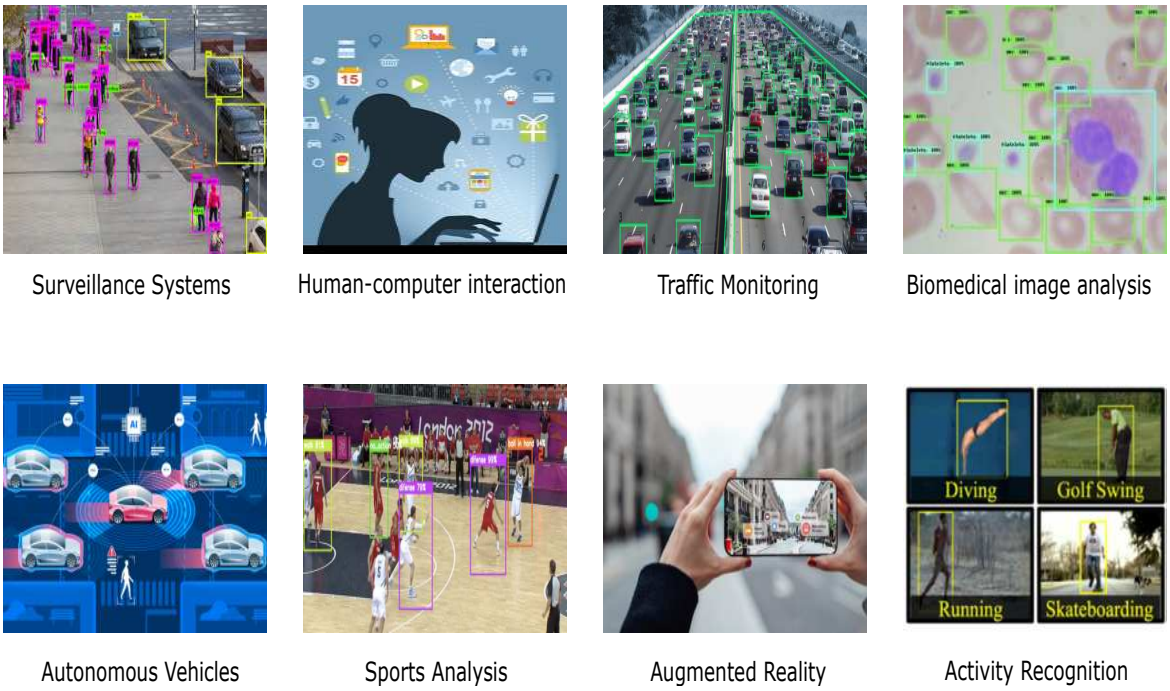


Fig. 1.3 Real-life applications of visual object tracking

expensive, and of higher quality than ever. Meanwhile, computing power has skyrocketed, and it has become far more efficient. In recent years, multi-core processing, graphical processing units (GPUs), and accelerators such as tensor processing units (TPU) have propelled computing platforms toward parallelization. This hardware enables near-real-time implementations of computer vision for object tracking. Tremendous advancements in deep convolutional neural networks (CNN) and GPU computing capacity are the primary drivers of significant progress in computer vision-based object tracking. Figure 1.3 shows the applications of visual object tracking in real-time scenarios.

Some of the real-life applications of object tracking are:

1. **Automated Surveillance and Security systems:** Surveillance is an essential aspect of security and patrol operations. The massive volume of data involved makes it impossible to guarantee suspicious behavior analysis by human operators for lengthy periods. Various automatic surveillance systems have developed due to recent advancements in computer vision technology. Object tracking would be significantly more important in security and surveillance. In a video sequence, intelligent visual surveillance systems aid in detecting and tracking suspicious or unexpected activities. The visual surveillance system necessitates quick and reliable technologies for identifying and monitoring moving objects.

2. **Video indexing:** Video indexing, like book indexing, allows viewers to find and browse content conveniently. Object tracking algorithms record the whole temporal contribution of the object of interest and can thus be used to create video indices.

3. **Human-computer interaction (HCI):** This involves designing and using interfaces between people and computers. HCI research focuses on people's interaction with computers and develops technology that allows them to do so in new ways. Object tracking plays a vital role in HCI by incorporating capabilities to track hand gestures and eye gaze.

4. **Traffic monitoring:** Vision-based traffic systems have grown in popularity in recent years, both in traffic monitoring and autonomous vehicle management. Video cameras monitor traffic and are part of an intelligent transportation system (ITS). Vehicle recognition and route tracking are critical in maintaining a smart traffic system in modern cities. They may provide vital information about traffic conditions such as vehicle velocity distribution and density and detect possible traffic jams.

5. **Medical diagnostic systems:** Medical image processing software is becoming more significant in supporting doctors with diagnosis, treatment planning, and image-guided therapies. In biomedical research, fast, accurate, and robust tracking of motile targets, such as cells, protrusions, intracellular particles, or deformable anatomical structures, is critical. The large amounts of multidimensional image data created by fully automated optical microscopes make manual analyses extremely error-prone and time-consuming, necessitating the creation of repeatable and accurate image analysis algorithms.

6. **Autonomous vehicles:** Autonomous/self-driving automobiles have stimulated academic and industry attention in recent years. An autonomous vehicle must comprehend the environment in which it is traveling. Real-time and accurate detection of all objects on the road is necessary to ensure the safe operation of vehicles. The autonomous vehicle must be able to locate itself in each environment and recognize and track objects (moving and stationary). The primary goal of autonomous driving is to detect and monitor vehicles, pedestrians, traffic lights, traffic signs, and other objects near the vehicle to assure safety.

7. **Sports analysis:** The quick and precise movements in most sports make the detailed observation and analysis difficult for the trainers and analysts. In situations like this, object tracking techniques can help bridge the gap between the sports event and analytical insights by using automated systems to find and monitor each player of interest throughout the film. Computer vision systems can now discriminate between

the ground, players, and other foreground objects, follow moving players, and identify the ball using object tracking algorithms.

8. **Augmented reality (AR):** Real-time 360° augmented reality experiences surrounding tangible objects are possible through object tracking. Scene tracking helps to augment scenes, rooms, and larger objects. Object tracking is an excellent AR feature for enhancing various products like industrial machines, historical monuments, etc. Camera posture can be determined using object tracking to align the virtual object with the camera view.

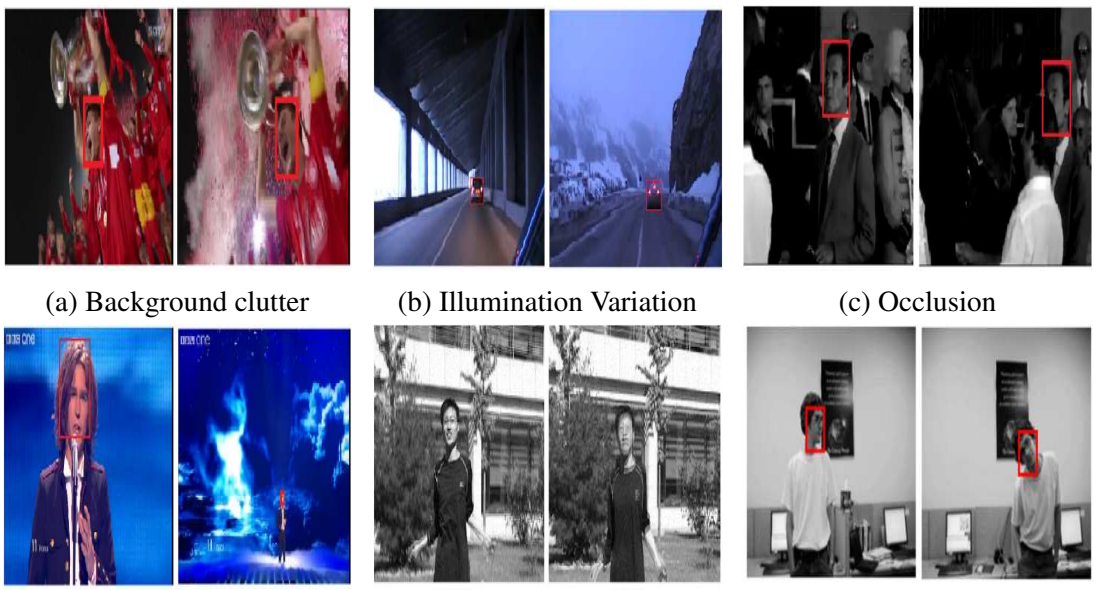
9. **Activity Recognition:** The challenge of employing computer vision to track and comprehend human behavior is significant. Human security officers are responsible for image interpretation and risk detection despite surveillance cameras installed. This observation assignment is not well suited because it requires prolonged concentration. Thus, there is a firm intention to build intelligent vision-based monitoring systems to assist human users in risk detection and analysis. Object tracking forms the foremost step in activity recognition.

1.1.3 Challenges in Single Object Tracking

In the tracking scenario, the target object in a given video does not change throughout the sequence, but the appearance and surroundings of the object may change continually. These changes cause the object's appearance to shift drastically compared to the first frame, making it harder to locate it in future frames. The object's appearance can change due to deformations or rotations, changes in camera motion that generate motion blur and scale changes, or environmental factors such as altering light conditions and shadows. An efficient tracking algorithm should be able to consistently track the object for the long-term without fail under all these situations.

1. **Background Clutter:** The accuracy of object tracking models depends on the backgrounds of inputted images or images used to train them. The background can occasionally be busy or surrounded by other objects when tracking an object, as shown in Figure 1.4a. It is easier for a tracking system to detect and track objects against a blurry or single-color background than against overly busy backgrounds, of the same color as the object, or congested.

2. **Illumination Variation:** The tracked object may be subject to various background illuminations, as shown in Figure 1.4b, depending upon the number and position



(a) Background clutter (b) Illumination Variation (c) Occlusion

(d) Scale Variation (e) Motion Blur (f) Object Rotations

Fig. 1.4 Challenges in single object tracking

of light sources used. However, when the target objects suffer severe appearance modification due to high illumination variation, the extracted target features will be insufficient to distinguish them from the background, causing the tracking algorithm to lose the target in the scene.

3. **Occlusion:** Occlusions profoundly impact the performance of object tracking systems. The target object is occluded if masked by another object in the current image, as seen in Figure 1.4c. Occlusions may occur when one part of the target blocks another (self-occlusion), when two tracked objects occlude each other (inter-object occlusion) or when a background structure occludes the tracked objects (background occlusion).

4. **Scale Variation:** Significant variations in size, shape, and intensity profile occur as objects move closer or farther away from the camera, as shown in Figure 1.4d. As the size of the search region significantly connects with the target scale, changing the target scale frequently influences position estimates.

5. **Motion Blur:** Motion blur induced by camera shake and object movement degrades visual perception quality and can also make video analysis jobs challenging to complete. The target may appear blurred due to the object or camera moving, as seen in Figure 1.4e, which may significantly impact the appearance model of the tracking algorithm.

6. **Object Rotations:** As demonstrated in Figure 1.4f, in-plane and out-of-plane rotations of the object create variations in the target appearance. As a result, the bounding box must orient with the target object. In real-world applications, recognizing accurate bounding boxes and predicting the object orientation is critical.

1.1.4 Classification of Object Tracking Algorithms

Due to the wide range of applications and various tracking challenges, VOT is an appealing study subject in the computer vision domain. Although the academic community has made significant achievements in recent decades, VOT still has a lot of potential to be explored further. VOT has stimulated the scientific community's interest, which has resulted in the development of several cutting-edge tracking algorithms. The existing object tracking algorithms can be classified based on several criteria.

1. **Detection-based and Detection free trackers:** Detection-based tracking complements tracker and detector in an integrated framework. Detection-based trackers work in two ways: In the first case, every frame is fed into a pre-trained object detector to generate object detections, further refined by the tracking algorithm. Here, the tracker deals with detection failures. Secondly, the object detector is executed only in selected frames, and the tracker is used to make the remaining predictions. The selective approach is well suited for long-term tracking. Detection-free tracking necessitates the manual initialization of objects in the first frame. In successive frames, the tracking algorithm localizes these objects. It fails to track new objects appearing in between the video sequence.

2. **Single object and Multiple object trackers:** Even when the environment contain multiple objects, single object trackers only track a single target. The initialization in the first frame defines the target object. Multiple object trackers locate all the objects in each frame throughout the entire sequence. It keeps track of the missed objects as well as any new object that appears in between.

3. **Online and Offline object trackers:** An offline tracker monitors an object in a recorded video. For example, offline trackers can track a player in a recorded game for strategic analysis. Offline trackers can generate more accurate tracking predictions by looking at past and future frames. Online trackers cannot use future frames to improve the results and are suitable when predictions are available instantly.

4. **Online learned and Offline learned object trackers:** Online-learned trackers often use the initiation frame and a few consecutive frames to learn about the object to track.

For example, to monitor a person wearing a red shirt at the airport, locate the individual in one or more frames using a rectangular bounding box. The tracker would learn the object appearance from these frames and monitors the object in the remaining frames. Unlike online learned trackers, offline learned trackers do not learn anything while running. These trackers learn the entire information offline. For example, a tracker trained to recognize people can be used to track all the people in a video in real-time.

5. **Generative and discriminative trackers:** The generative appearance models are primarily concerned with how to fit the data from the object class appropriately. In practice, however, verifying the validity of the described model is quite challenging. They incrementally learn visual representation for the foreground object region while disregarding the influence of the background by providing online-update techniques. Traditional online learning approaches perform object tracking by searching for regions most similar to the target model. The tracking framework incorporates an online learning technique to adaptively update the target's appearance model in response to changes in appearance. Object tracking becomes a classification problem using discriminative approaches. The trained classifier can be updated online and is used to distinguish the target from the background. This approach uses both target and background information at the same time. An online learned binary classifier discriminates the target from the background and is updated online to adapt to the target variations.

VOT has piqued the scientific community's interest, which has resulted in the development of many cutting-edge tracking algorithms. The thesis focuses on model-free, causal, and discriminative, correlation filter based single-target tracking. The term model-free indicates that the tracker is learned from the training samples derived from the given sequence. The causality property implies that the tracker locates the target in a given frame with no knowledge of the subsequent frames. A rectangular bounding box defines the output of trackers in each frame.

1.2 Correlation Filter based Object Tracking

Compared to state-of-the-art trackers, correlation-filter (CF) based discriminative trackers have recently acquired interest because of their simplicity, efficiency, and robustness. The correlation operation transforms to simple, element-wise multiplication in the Fourier domain, and hence correlation filters are computationally efficient. As a result, their popularity in visual tracking has increased. Correlation filters are initialized with a target patch cropped at

the target center from the first frame of the sequence. The target is localized in the subsequent frames based on the estimated position in the previous frame.

A suitable feature extraction approach generates a feature map from the input patch to represent the target's appearance adequately. A cosine filter smoothens the edges of the extracted feature map, and the correlation operation replaces the convolution procedure. The element-wise multiplication of the learned correlation filter and extracted features generates a response map, employing a Discrete Fourier Transform (DFT). Inverse Fast Fourier Transform (IFFT) is used over the response map to provide a confidence map in the spatial domain. The new target position is the location of the highest confidence score. Online feature and filter updates learn the target appearance at each new location.

The primary research focus on correlation filter trackers is as follows: (i) Improve learning methods; (ii) Improve the regression equation; (iii) Extract more powerful features; (iv) Reduce the influence of scale shift; (v) Weaken the impact of boundary effects; (vi) Use a more robust confidence criterion; (vii) Combine with the long-term target memory model, and so on.

1.3 Motivation and Research Objective

Correlation filter (CF) based trackers have recently aroused attention in visual object tracking and have shown impressive results in many competitions and benchmarks. Though numerous tracking algorithms based on correlation filters are available today, most fail to efficiently detect the object in an unconstrained environment with dynamically changing object appearance. Although correlation filter -based object tracking methods have an excellent real-time tracking performance, they still face several concerns, in addition to the challenges discussed in Section 1.1.3. Object deformations caused by rotation and scale changes, model drift issues, long-term tracking capabilities, and tracking resumption are among the top concerns in correlation filter based single object trackers. In view of these challenges, the proposed research aims to build "generic approaches towards enhancing correlation filter trackers."

On thorough analysis of the existing object tracking algorithms based on correlation filters, we have identified the following limitations of the correlation trackers, which form the basis of the proposed research works. The identified limitations of the correlation filter based tracking algorithms can be summarized as follows:

1. **Lack of adaptability to object rotations and illumination changes in the environment:** Despite all the advances in CF tracking, most correlation filter based tracking methods are still sensitive to object deformations, rotations, and changes in illumination. The scarcity of robust training features extracted from the preceding frames

limits the learned appearance model's capacity to adapt to changes in the target item. As a result, the impact of oriented bounding boxes over axis-aligned bounding boxes in each frame of a video sequence is studied and rotation adaptiveness is incorporated to extract better features from each frame that helps in learning a robust appearance model. Illumination correction strategies are also presented for extracting advanced information from prior frames, which aid in learning a robust appearance model.

2. **Absence of tracking resumption in correlation filter trackers:** Accurate and robust visual object tracking is one of the most challenging computer vision problems. Recently, discriminative correlation filter trackers have shown promising results on benchmark datasets, continuously improving tracking accuracy and robustness. Still, these algorithms fail to track as the target object and background conditions undergo drastic changes over time. They are also incapable of resuming tracking once the target is lost, limiting the long-term ability to track. The proposed research investigates the need for a generic technique to identify tracking failures in correlation filter trackers and learns a mechanism to re-initialize the tracker upon a failure in any frame.

3. **Model-drift in correlation filter trackers:** Object tracking relies on a recursive search technique around the previous target location, concurrently learning the target appearance in each frame. A failure in any frame causes a drift from its optimal target path. The incorrect background information from loosely defined bounding boxes also misleads the appearance model learned from each frame. Analysis of the model drift problem in correlation filter trackers has proven the necessity of obtaining highly confident search regions in each frame. Consequently, the benefits of leveraging object segmentation as a guiding step in tracking is identified.

4. **The limited ability to co-localize objects in consecutive frames:** The model drift and online model update from incorrectly tracked frames reduce the localization capability of correlation filters. Thus, locating common objects between successive frames is beneficial considering these limitations. Moreover, the advantage of segmented object masks over bounding boxes in target representation is recognized. These factors have motivated the usage of co-attention maps for target localization in the tracking scenario.

1.4 Research Contributions

The thesis proposes generic approaches to enhance correlation filter trackers, given the limitations of the existing correlation filter trackers. The proposed methods are "generic"

because they are integrable on any correlation filter tracker as the base. The proposed methods are built on the competing state-of-the-art trackers of the corresponding times and compared with the state-of-the-art trackers on various benchmark datasets. The initial study on correlation filter-based trackers incorporates rotation and illumination invariance into the correlation tracker framework. Further investigations lead to the development of three generic approaches to enhance the accuracy and robustness of correlation filter trackers. The proposed approaches generate more confident search regions and incorporate target-specific attention into correlation filter trackers, enabling better resumption, improved tracking ability, and reduced failure rates. Compared to the recent state-of-the-art trackers, the proposed trackers provide competing performance with significant improvements over the baseline algorithms in accuracy and robustness.

The major contributions of the thesis can be summarized as:

1. **Learning rotation adaptive correlation filters for robust visual object tracking**

    - An Illumination Correction (IC) Filter introduced in the tracking framework eliminates the adverse effects of variable illuminations on feature extraction.
    - The proposed approach incorporates rotation adaptiveness in standard CF by optimizing the orientations of the target object in the detector stage.
    - The orientation optimization helps extract robust features from correctly oriented bounding boxes, unlike most state-of-the-art trackers that rely on axis-aligned bounding boxes.
    - The sub-grid localization cost function is supervised in the detector stage of CF trackers. This cost function intends to eliminate the false positives during detection.
    - The impact of enhancing smoothness through displacement correction is analysed.

2. **Detection based long-term tracking in correlation filter based trackers**

    - A generic technique adaptable to any correlation tracker is proposed to incorporate tracking resumption.
    - An adaptive threshold decides the tracking uncertainties in correlation trackers.
    - A detector is formulated in the tracking scenario using Bag-of-Visual-Words (BoVW).
    - A tracker re-initialization scheme eliminates the model drift.

- An online model update adapts to the target variations and a similarity matching technique estimates the final target location.

3. **Domain adaptive-segmentation guided attention for correlation filter tracking**
    - Addresses segmentation guided attention mechanism for single object tracking proposed using a region proposal module (RPM) and target localization module (TLM) driven by a correlation filter tracker.
    - Two generic variants of the proposed segmentation guided tracking are developed: i) SACOT and ii) DA-SACOT (without and with domain adaptation, respectively).
    - The region proposal module utilizes an initial offline training to carefully integrate domain-specific knowledge into the tracking pipeline. The localization module learns the target appearance through an online model update.
4. **Co-Attention Maps for Discriminative Object Tracking (CAM-DOT)**
    - Proposes the use of co-attention maps to generate confident search regions.
    - A Siamese encoder-decoder network generates co-attention maps.
    - Enhances the target localization using the pre-trained segmentation network by replacing the inputs with the proposed co-attention maps.
    - Generates segmented object masks and bounding boxes as output target representations in each frame.

1.5 Organization of the thesis

This thesis presents generic approaches to enhance the accuracy and robustness of correlation filter trackers by incorporating adaptiveness to various object deformations and environmental changes. This chapter outlines the research theme. Chapter 2 details the generic framework of correlation filter-based tracking, focusing on the existing research explorations in correlation filter-based trackers. A summary of the tracking benchmark datasets and evaluation protocols used for the experimental analysis of single object tracking algorithms is discussed in Chapter 3. Chapter 4 elaborates a generic technique for incorporating rotation and illumination adaptiveness in correlation filter trackers. The proposed tracking framework supervises the detection stage of discriminative correlation filter trackers by eliminating false positives in the convolution response map. Further, the impact of displacement consistency on CF trackers is demonstrated. The generic nature and efficiency of the proposed framework is illustrated by integrating the contributions into two state-of-the-art CF trackers: SRDCF

[2] and ECO [10]. Chapter 5 details a detection-based long-term tracking approach to incorporate tracking resumption in correlation filter trackers. The method introduces tracking resumption in correlation filter based single object trackers. A detector mechanism re-initializes the tracker upon a target loss identified using a variable and adaptive threshold. Online update of both the tracker and detector modules incorporates temporal information into the proposed framework, making it robust to object appearance changes. The tracker and detector stages correct the false appearances learned from any frame, complementing each other, and reducing the model-drift issue. A similarity matching scheme estimates the final target location. Chapter 6 introduces instance segmentation as an attention mechanism in the object tracking framework. The proposed tracking framework employs a region proposal module (RPM) based on instance segmentation to search for region proposals having a high probability of being the target. Using a correlation filter, a target localization module (TLM) localizes the final target. A domain adaptation technique in RPM and TLM modules incorporates target-specific knowledge and strong discrimination ability. Chapter 7 proposes an object tracking algorithm that uses an end-to-end trained convolutional neural network (CNN) that outputs co-attention maps that highlight the locations of common objects between consecutive frames. A target localization module based on a correlation filter follows the CNN module. The proposed tracker generates segmented object masks and rectangular bounding boxes to represent the target object in each video frame. Chapter 8 summarizes the salient aspects of the exploration and assessment of correlation filter-based object trackers and the significant contributions of the thesis, with some relevant remarks on possible future research extensions.

Chapter 2

Correlation Filter based Tracking Algorithms

Visual Object Tracking (VOT) is an exciting yet challenging computer vision subfield. Although the research community has made remarkable achievements in recent decades, VOT still has much potential to be explored further. The tracking difficulties have prompted the development of several techniques. As a result, there exists a significant need to compile the literature on these issues, assess the resiliency of the trackers, and classify these algorithms in accordance with the issues raised by the benchmarks that are now in use.

Tracking algorithms come in two types: generative and discriminative. The generative models track by looking for the best-matched window whereas the discriminative models distinguish the target patch from the backdrop. Recent tracking algorithms are split into two groups in the proposed study: CFTs (Correlation-Filter-Based Trackers) and Non-CFTs (Non-Correlation-Filter-Based Trackers) (NCFTs). Discriminative tracking algorithms have made substantial progress in differentiating the object from the background. This chapter provides an orderly reference for the several correlation filter trackers currently in use, to point out areas for future research, and to offer suggestions for creating new tracking algorithms.

2.1 The standard correlation filter tracker framework

To reduce computational costs, CF-based tracking systems compute in the frequency domain. The general design of these algorithms is shown in Figure 2.1 and follows the "tracking-by-detection" method. Initialize correlation filters using a target patch from the first frame of the sequence cropped at the target location. Based on the target's estimated position in the previous frame, the tracker estimates the target's location in the subsequent frame. A feature

map reflecting the target's appearance is generated from the input patch using a suitable feature extraction method. A cosine filter smoothens out the edges. Instead of using the exhausted convolution procedure, the correlation operation is used.

The response map is created by multiplying the adaptive learning filter and retrieved features element-wise and employing a Discrete Fourier Transform (DFT). DFT uses the Fast Fourier Transform to operate in the frequency domain (FFT). Inverse FFT (IFFT) is used in the response map to provide a confidence map in the spatial domain. The new target position is estimated using the highest confidence score. By extracting features and changing correlation filters, the target appearance at the newly predicted position is updated.

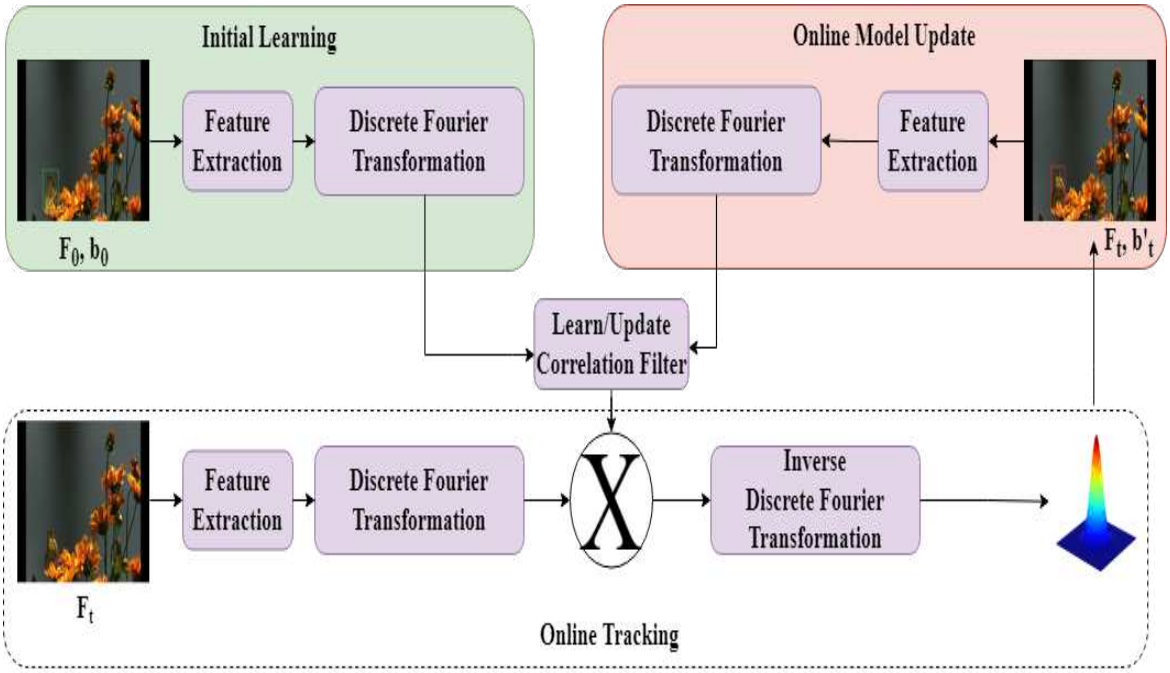


Fig. 2.1 The generic working of a correlation filter tracking algorithm

Let h stand for a correlation filter and x for the current frame, which could be the extracted features or the raw image pixels. Correlation in the frequency domain computes a response map by conducting element-wise multiplication between zero-padded versions of $f(h)$ and $f(x)$ and is identical to circulant convolution in the spatial domain, according to the convolution theorem. Because h is often much smaller than x, zero padding is employed before the Fourier domain transformation to ensure that the modified sizes of both are the same.

$$x \otimes h = F^{-1}(\hat{x} \odot \hat{h^*}) \tag{2.1}$$

where F^{-1} stands for IFFT, ˆ for Fourier representation, $\otimes$ symbolises convolution, $\odot$ implies element-wise multiplication, and $*$ is the complex conjugate. A confidence map between x and h is produced by the equation. The estimated target around the maximum confidence position is chosen to update the correlation filter. Assume the desired output is y. For new target appearance z, correlation filter h must meet.

$$y = F^{-1}(\hat{z} \odot \hat{h^*}), \hat{h^*} = \frac{\hat{y}}{\hat{z}} \tag{2.2}$$

The intended output y in the frequency domain is denoted by $\hat{y}$, and the division operation is executed during element-wise multiplication. Because circulant convolution has a complexity of $O(n^4)$ for image size $n \times n$, FFT minimizes the computational cost to $O(n^2 logn)$.

Training the intended appearance (orientation, shape), which may change over time, is challenging for CF-based tracking frameworks. Another problem is deciding on an efficient feature representation because powerful features can help CFTs perform better. Scale adaptation is another significant difficulty for CFTs, as the size of correlation filters is fixed during tracking. The magnitude of a target might alter over time. Furthermore, once the object is lost, it cannot be retrieved.

2.2 Correlation filter based object tracking: the literature review

As mentioned below, CFTs are further classified into Basic CFTs, regularised CFTs, part-based, Siamese-based, and Fusion-based CFTs.

1. **Basic Correlation Filter based Trackers:** Kernelized Correlation Filters (KCF) are used as the baseline tracker by basic-CFTs. Different features, such as the Histogram of Gradients (HOG), Color Names (CN) [11], and deep features from neural networks [12], and Residual Networks [13], are provided to trackers. [14–27] are just a few trackers constructed utilizing KCF as the basis tracker. KCF is a tracking algorithm that distinguishes between the target object and its background using a Gaussian kernel function. Four-cell HOG descriptors are used by KCF. The target position is predicted by the confidence map with the highest confidence score that was created by using the inverse Fourier transform on the response map. The correlation filter is updated by cropping a new patch to include the object and recalculating the HOG features. [28] uses extensive hierarchical convolutional features. Each subsequent frame's cropping of the search area centres it on the previously determined target position. Three hierarchical convolutional features are used to exploit the target's appearance.

Deep features are resized using bilinear interpolation to the same size. Each CNN feature's response maps are created using a separate adaptive CF. The new target position is estimated using the correlation response maps and a coarse-to-fine approach. Adaptive hierarchical CFs are updated on the newly anticipated target location. The Hedged Deep Tracking (HDT) [29] algorithm employs multiple CNN feature levels. To produce a single robust tracker, the authors of HDT integrated several weak trackers. During tracking, a new image is cropped and six deep features are computed with VGGNet using the target position from the previous frame. To create response maps for each CF, deep features are used. The target position is estimated by each weak tracker. A random fern classifier [30] is used to re-detect the target online throughout tracking in Long-term Correlation Tracking (LCT) [20], which employs correlation filters to estimate target translation and scale only. In the event of a failure, the LCT tracking algorithm does re-detection. The online random fern classifier is used to re-detect the target if the predicted target score is less than a predetermined threshold. The posteriors of all the ferns are used to calculate the average reaction. The Multi-task Correlation Particle Filter (MCPF), which makes use of a particle filter structure, was proposed in [31]. To guide particles into the search region, the MCPF makes advantage of all target states. The target position is computed as the weighted sum of the particle response maps produced by the MCPF. Discriminative Scale Space Tracking (DSST) [16] estimates translation and scale independently utilising independent CFs. To estimate scale, the target sample is learned across many scale changes. It predicts the initial translation by applying a standard translation filter to each input frame. [14] used two separate regression models, one for each target representation, to address the tacking problem. The tracking design trains a model using two complementing features from two different patch images. The target representation is made up of global colour histograms and HOG features. Each incoming frame has a search region centred at the previously calculated position, and its HOG properties are convolved with CF to provide a dense template response. Using a linear combination of template and histogram response scores, the target position is determined. By scanning patches at various scales, [22] modifies the target appearance and scale estimation using residual learning. Relative and base mapping are used to calculate the response map. By looking at different scaled patches at the newly determined target centre position, the scale is estimated. The Multi-Store tracker [18] aggregates visual information using short- and long-term stores to prevent drifting and stabilise tracking. The long-term storage manages the output using RANSAC estimation and key point matching and short-term storage uses an integrated correlation filter.

2. **Regularized Correlation Filter Trackers (R-CFTs):** Because filter and patch sizes must be equal, discriminative CF (DCF) tracking methods have a limited detection range. The DCF might catch up on the background for irregular objects. The centre of DCF response maps contains accurate scores, while other values are periodically influenced by assumptions, which degrades DCF performance. The fact that DCFs can only search in a limited region is another disadvantage. DCF trackers perform badly on a target deformation problem due to the overfitting of the model produced by learning from target training data but omitting the negative examples. As a result, the tracker is unable to re-detect in the event of occlusion. The occlusion issue might be resolved by expanding the search area, but the model will pick up background knowledge and lose its ability to discriminate. In order to solve these DCF constraints, regularisation is necessary, and these trackers are known as Regularized Correlation Filter Trackers. [32, 33, 10, 34, 2, 35–41] are some of the R-CFTs available in the literature.

   Spatial regularisation is introduced in DCF learning by Spatially Regularized DCF [2]. The background information during tracking is weakened by the regularisation component. Spatial regularisation limits the filter coefficients based on spatial data. Giving coefficients outside the targeted region higher values, and vice versa, suppresses the background. The SRDCF framework has been enhanced with deep CNN features in [34]. The SRDCF framework has been expanded in [35] to include degraded training samples. While estimating high-quality samples, it lessens the weight of degraded training samples. It uses training samples from earlier frames to identify the patches that require correction and gives them heavier weights. To produce temporal regularisation, passive-aggressive learning is integrated into the SRDCF using a single image [40]. Deep Motion SRDCF (DMSRDCF) [37] combined deep motion data with conventional features using SRDCF as a baseline tracker. [36] learns multi-resolution feature maps for tracking. Response maps are produced as a result of the convolutional filters being trained at various resolutions. The many response maps are then combined to create a single, final response map that can be used to infer the target's position. A Gaussian Mixture Model is used by ECO [10] to represent different target appearances and reduce the number of filters needed to effectively capture target representation by matrix factorization. DCF is used in conjunction with channel and spatial reliability in [41]. By building a spatial binary map to only learn target information at the present target position, this assures spatial consistency. The foreground and background models, which are represented as colour histograms, are used to calculate the appearance likelihood using the Bayes' rule. In order to calculate high responses for targets and low responses for context information, [42] combines

global context information into a baseline tracker. Background patches are used in the Background Aware Correlation Filter (BACF) tracker [33]. In order to take advantage of target dynamics, the Attentional CF Network (ACFN) [32] employs an attentional mechanism. An ACFN is composed of an Attentional Network and a CF Network. Within the subset of selected tracking modules, the tracking module with the best reaction is the target.

3. **Siamese-Based Correlation Filter Trackers:** Two inputs are combined into a single output by a Siamese network. The objective is to check the two image patches sent into the network for any instances of identical objects. The network examines the two inputs to determine how similar they are and may simultaneously learn about similarity and features. Initially, the Siamese network concept was used to verify signatures and identify fingerprints. The Siamese architecture makes use of CNN and measures the similarity of two images using layers that are shared. CFTs that have been combined with a Siamese network for visual tracking and used to address tracking difficulties are known as Siamese-Based CFTs [43–47].

   A fully convolutional Siamese network is proposed in [43] that attack the tracking problem through similarity learning. They compare a target image with a similar size candidate image and provide high scores if the two images are the same. An example target and a search patch that is bigger than the target predicted in the previous frame are used as inputs, and the output is a scalar-valued score map. The network makes use of a correlation layer and a convolutional embedding function to combine the deep feature maps of the target and search patches. The final target position is approximated using the highest value in the response map. The Correlation Filter Network (CFNet) in [46] performs end-to-end learning of underlying feature representations via gradient backpropagation. The primary tracker for online tracking is [43]. Based on the previously estimated target location, target features are compared with the wider search region on a new frame. A similarity map between the search patch and the target template is produced by the cross-correlation. [47] uses an offline learned light weight network with correlation filters. The CF layer is adapted through backpropagation of the probability map of the target estimate. Regularized linear regression is used to learn target appearance and background suppression from previous frames in [44]. The search patch is multiplied with the Gaussian weight map that was learned from the first frame to the current frame to accomplish background suppression.

4. **Part-Based Correlation Filter Trackers:** While most CFTs learn the target template in its entirety, these trackers learn the target appearance in chunks. The problems of

targets being obscured or deformed in sequences are addressed by several part-based trackers. [48–54].

Each component of the object is given a spatial constraint by [53] and are tracked separately using the KCF tracker. During tracking, the confidence score map for each component is generated by applying adaptive weights to each new input frame. The confidence score maps are combined into a single map by adding adaptive weights, and a new target position is determined by the particle filter method. Parts with weights higher than a threshold value are updated in adaptive parts-based trackers. In [51], a particle filter architecture is presented that makes use of the local context by using reliable patches to monitor the target and employs KCF as each particle's basis tracker. Each trustworthy patch's weight is determined during tracking depending on its trackability and whether it possesses the desired features. A patch is deleted and re-sampled to estimate a new reliable patch if it is no longer reliable. By gathering all the weighted, trackable, and reliable positive patches, a new target position is predicted. RNN-based confidence maps weigh the adaptive CFs in [50] to minimize the detrimental effects of background clutter. [49] is a particle filter system that uses KCF as the basis tracker to train target patches. Both holistic and object components data are utilised by the structural correlation tracker in [48]. The target location is estimated using weighted responses from unobstructed regions. Structural CF was created by [52] to take advantage of an object's spatial arrangement among its parts in its dual form. The position of each component is estimated using the maximum response of the response map. The weighted average of all translations is then used to calculate the target location.

5. **Fusion-based Correlation Filter Trackers:** [55–59] are only a few of the trackers that employ various types of fusion. [56] used visual and infrared images for pixel level fusion, with HOG, and grey values, whereas Colornames were used for feature level fusion. [57] proposed a feature fused tracker employing raw pixels, color histogram, and Haar characteristics. The region of interest was divided into sub-parts and the features were evaluated individually and then merged using weighted entropy. [59] proposed a deep feature fusion method based on a Local Detection Network and the Global Network Detection Network. If the confidence score falls below a threshold, the target is declared lost. [58] developed Correlation-based Tracker level Fusion by combining two complementing trackers, TLD [60] and KCF [61]. If the output of the TLD tracker is valid, the output of the KCF tracker is computed during tracking. Conservative correspondence is determined for both trackers as a confidence score

based on the first 50% of positive patches resembling the sample patches. The tracker output with the highest score is the current bounding box.

2.3 Chapter Summary

Discriminative correlation filters (DCF) based visual tracking algorithms have gained popularity in recent years because to their exceptional performance and quickness. Correlation filters use a fast Fourier transform to convert spatial correlation into element-wise operation in the frequency domain. However, improved DCF trackers suffer from a high number of calculations or an inadequate model updating strategy, which prevents them from meeting the demands of target tracking when the computing resources are strictly constrained. Traditional DCF trackers, like those mentioned above, are also impacted by boundary effects. The several types of correlation filter-based tracking algorithms are compiled in this chapter. Each algorithm offers a unique approach for utilising target structure to predict target location in a sequence.

Chapter 3

Benchmark Datasets and Evaluation Protocols

Object tracking is still a challenging problem despite being explored for decades and significant progress achieved. Various parameters, including illumination variation, occlusion, and background clutter, influence the performance of a tracking algorithm. As a result, it is critical to properly evaluate tracking algorithms to highlight their strengths and weaknesses and recommend future research directions. It is necessary to acquire a sample dataset for a complete performance review. The OTB [62], VOT [1], UAV [63], and TempleColor [64] datasets are a few of the datasets available for single object tracking. Ground-truth annotations and attributes (e.g., occlusion, fast motion, or illumination change) are included in these benchmark datasets further to study the performance of the evaluated tracking methods. This chapter details the evaluation protocols, video sequence analysis, and attribute - based annotations of the benchmark datasets used in the proposed works.

3.1 The Visual Object Tracking (VOT) Dataset

The VOT challenges give the visual tracking community a well-defined and reproducible method of comparing trackers and a shared forum for discussing visual tracking evaluation and developments. The challenge aims to create a database of significant benchmarks and host workshops or other similar activities to advance visual tracking research. The Visual Object Tracking challenge compares single-object trackers that do not use pre-learned object appearance models across short periods. The VOT is the most extensive and most difficult tracking benchmark because of the enormous number of tested state-of-the-art trackers. The VOT challenge has pushed single-object tracking toward performance evaluation standards

since 2013. The primary purpose of VOT is to create datasets, evaluation measures, toolkits, and a platform for debating evaluation-related issues through organizing tracking challenges.

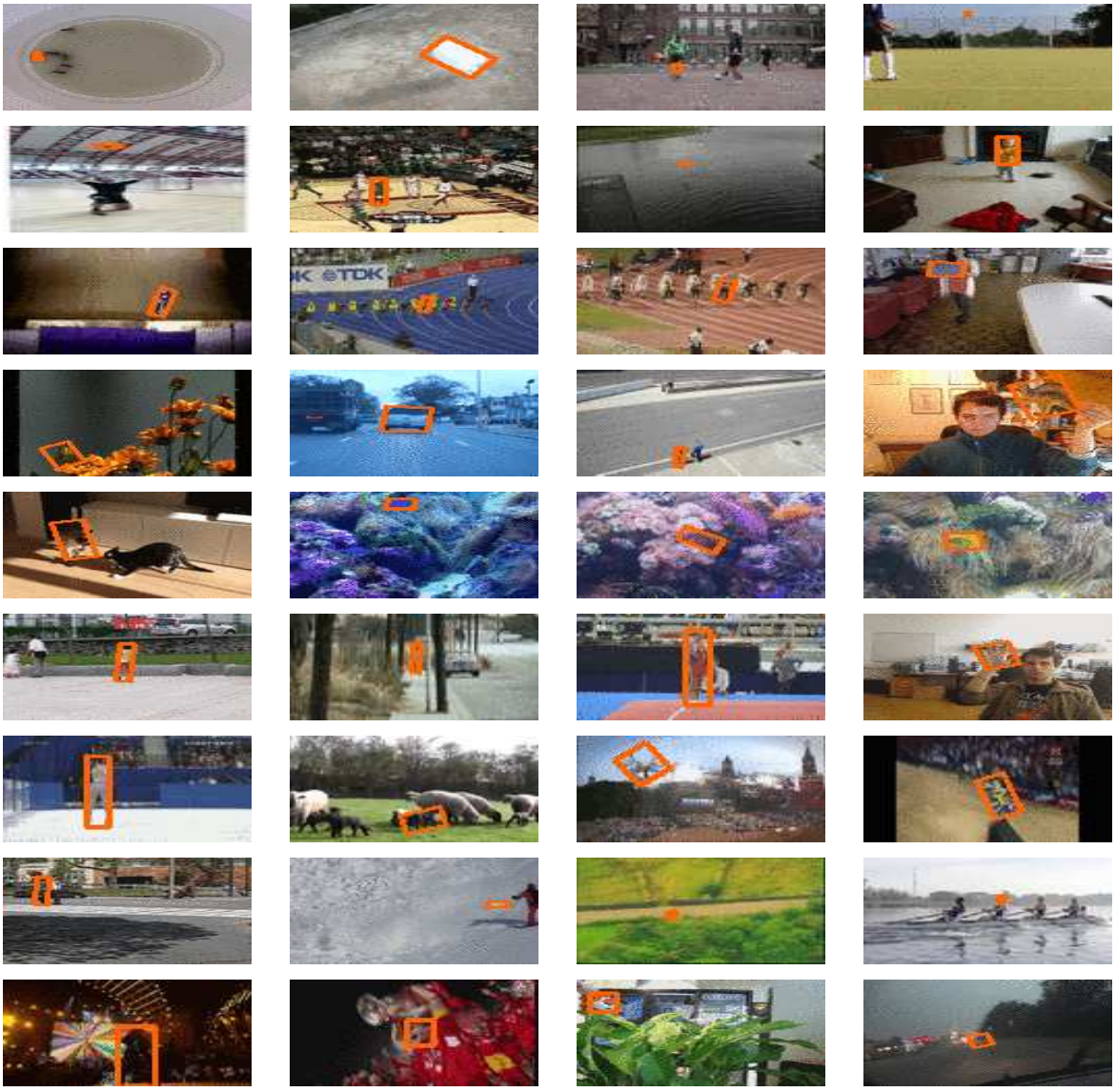


Fig. 3.1 Initial frames of sample sequences from the VOT dataset with the target object depicted in a rectangular bounding box.

Recent performance evaluation methodologies use the intersection over union (IoU) between the ground truth bounding box and tracker prediction bounding box to measure tracking accuracy. The performance analysis of the VOT reset-based protocols uses two metrics: accuracy and robustness. They count the number of times the tracker fails and how much overlap during successful tracking periods. The expected average overlap (EAO), a logical combination of accuracy and robustness, has been the primary metric of the VOT

since 2015. The VOT reports the so-called state-of-the-art bound (SotA bound) on its annual challenges. Any tracker that exceeds the SotA bound is considered state-of-the-art by the VOT standard. This constraint counters the habit of only considering trackers that rank first on benchmarks as state-of-the-art. The goal of SotA bound was to eliminate the necessity for fine-tuning benchmarks and encourage community-wide investigation of a more extensive range of trackers rather than just attaining the top rank. In VOT2014, tracking speed was identified as a significant parameter but was later dropped due to the limited normalizing capability and because speed varies a lot throughout the track.

The VOT has created a dataset construction process for building relatively big and demanding datasets from a vast pool of sequences throughout the years. The VOT2017 competition implemented a sequestered dataset examination for the primary short-term challenge winner determination. The VOT dataset contains 60 video sequences made public, and 60 are kept private. Solely the first dataset was made public, while the second was kept confidential and was only used to determine the winner of the main VOT2018 short-term challenge. A rotational bounding box annotates the target in the sequences, and the following visual properties label each frame of the video sequences: (i) occlusion, (ii) illumination change, (iii) motion change, (iv) scale change, and (v) camera motion. Unassigned frames form a sixth category that does not relate to any five attributes. Figure 3.1 shows sample sequences from the VOT benchmark dataset.

The VOT challenges employ a reset-based methodology and use accuracy (A), robustness (R), and expected average overlap (EAO) as the three metrics to evaluate the tracking performance. A failure occurs when a tracking algorithm predicts a bounding box with no overlap with the ground truth, and the tracker is re-initialized five frames later. The primary measures utilized to examine tracker performance in the reset-based tests are accuracy and robustness. The tracking accuracy is computed as the average overlap between the predicted and ground truth bounding boxes. The robustness of the tracker is measured by how many times it loses the target (fails) while tracking. Since the accuracy metric ignores ten frames after re-initialization, the possible bias due to resets decreases. For stochastic trackers run 15 times, average accuracy and failure rates represent the performance. The expected average overlap (EAO), the third and most important measure, predicts the average overlap on an extensive collection of short-term sequences with the same visual features as the given dataset. Due to changing sequence lengths, the measure tackles the problem of increasing variance and bias in the average overlap (AO) measure.

The toolkit uses a reset-based VOT evaluation procedure when the tracker bounding box does not overlap with the ground truth, which resets the tracker. VOT frame skipping reduces the correlation between resets.

3.2 Object Tracking Benchmark (OTB) dataset

Recently the availability of object tracking codes for evaluation has significantly improved. These tracking methods support different input formats (e.g., avi videos or raw image sequences), motion models (e.g., 2D translation, similarity, or affine transforms), and diverse output formats. This diversity necessitates the incorporation of these algorithms into a library for evaluation on a common platform. The object tracking benchmark incorporates the popular publicly accessible trackers in a code library with uniform input and output formats for performance evaluation. Furthermore, it provides a large benchmark dataset with ground-truth annotations and properties (e.g., occlusion, fast motion, or illumination change) to better analyse the performance of the evaluated tracking systems.

One significant issue when evaluating tracking algorithms is that the given results rely on a few sequences with varying initializations or parameters. Because an object detector may be employed to locate the item in the first frame, inaccurate target localization is typical. In addition, an object detector can be utilized to re-initialize the tracker after a tracking failure. OTB proposes perturbing the initial object states spatially and temporally based on ground-truth target locations for a fair and thorough evaluation. This evaluation methodology better examines the sensitivity of a tracking algorithm to initialization (i.e., robustness).

Several benchmark datasets for diverse vision problems have been produced, including the Berkeley segmentation [65], FERET face recognition [66], and optical flow dataset [67]. The VIVID [68] and CAVIAR [69] databases are two examples of benchmark datasets for tracking in surveillance scenarios. Most image sequences lack accurate ground-truth annotations since the trackers are not initialized and evaluated on the same platform. Hence, the published quantitative results in the literature are inconsistent. OTB gathered and annotated the most regularly used tracking sequences to provide a fair performance evaluation. The tracking benchmark TB-100 dataset has 100 target objects. Since some of the target objects are similar or less challenging, the OTB50 dataset created for an in-depth examination has 50 difficult and representative ones from the OTB100 dataset. Because humans are the most common target objects in practice, the TB-100 dataset comprises more sequences of this type (36 body and 26 face/head videos) than the other types.

3.2.1 Attributes of a test sequence:

Many factors influence the experimental results, making it challenging to evaluate tracking algorithms. The sequences are grouped according to the 11 attributes to better study the strengths and limitations of tracking algorithms. Each attribute represents a distinct challenge to object tracking. Many attributes can be attached to a sequence, and some appear more

frequently than others. OTB reports tracking results of sequences with specific attributes and the performance evaluation on the TB-100 dataset. Sequences with the same attributes can help understand the characteristics of tracking algorithms. For example, 49 sequences (29 in TB-50) labelled with the *OCC* property can aid in testing how well the tracker handles occlusion. The first frames of selected targets of the OTB dataset are shown in the Figure 3.2, along with ground-truth bounding boxes.

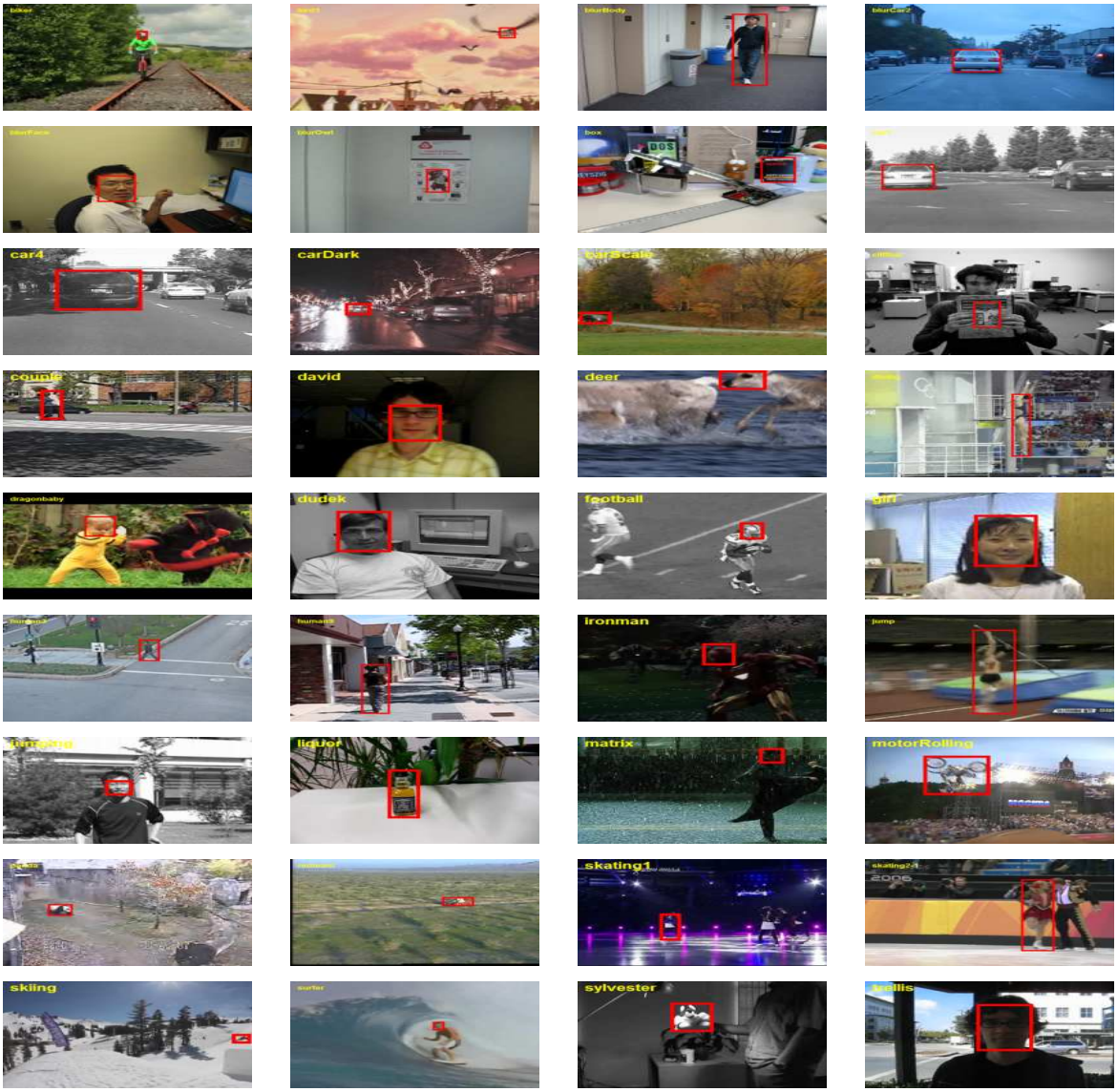


Fig. 3.2 Initial frames of sample sequences from the OTB dataset with the target object depicted in a rectangular bounding box.

3.2.2 Evaluation Methodology:

Assessing the performance of a tracking algorithm using quantitative metrics is difficult. Many parameters can be considered, including location accuracy, robustness against specific appearance changes, tracking speed, memory requirements, and ease of usage. Numerous metrics quantify accuracy in a single frame with the tracking output and ground-truth object state. When an algorithm loses track of a target object, it may resume tracking after a failure if a re-detection module exists or locates the target object again as it reappears at the tracking bounding box's location. The assessment might not be fair in averaging the evaluated values of all frames in a sequence because a tracker may have lost the target initially but might have tracked it successfully if initialized in an object state or frame.

Precision plot:

The center location error, which computes the average Euclidean distance between the tracked targets' center locations and the manually labelled ground truth positions of all the frames, is a widely used assessment statistic for object tracking. When a tracking algorithm loses track of a target object, the output position can be arbitrary, and the average error value thus fails to reflect tracking performance accurately. Instead, a better metric to quantify tracking performance is the percentage of frames in which the estimated locations are within a set threshold distance of the ground-truth positions. On the other hand, the center location error merely measures the pixel difference and does not account for the target object's size and scale.

Success plot:

The overlap and average overlap scores (AOS) are other commonly used evaluation metrics. The overlap score is defined as $S = \frac{r_t \bigcap r_0}{r_t \bigcup r_0}$, where $\cap$ and $\cup$ represent the intersection and union operators, respectively, and $|.|$ specifies the number of pixels in a region, given a tracked bounding box r_t and the ground-truth bounding box r_0 of a target object. The overlap score determines whether an algorithm effectively tracks a target object in a single frame by comparing S to a specific threshold t_0. The success rate changes as the threshold vary between 0 and 1. In addition, performance evaluation utilizes the average success rate with a predefined threshold of $t_0 = 0.5$. The area under curve (AUC) of each success plot, the average of the success rates corresponding to the sampled overlap thresholds, is another metric for ranking trackers.

3.2.3 Robustness Evaluation

The most typical way of evaluating an algorithm is the One-Pass Evaluation (OPE) which initializes the tracker with the ground-truth object state in the first frame and then averages the precision or success rates. This metric has two fundamental downsides, despite its simplicity. To begin with, a tracking algorithm may be sensitive to initialization in the first frame, and its performance with various initial states or frames may change dramatically. Second, most algorithms lack re-initialization techniques. Thus, the tracking data after tracking failures are meaningless. By perturbing object states temporally (i.e., starting at different frames) or spatially (i.e., starting with different bounding boxes), OTB provides two metrics to determine if a tracking method is robust to other object states. The terms temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE) describe these evaluation metrics.

Temporal Robustness Evaluation (TRE)

Each tracking algorithm is evaluated multiple times from various starting frames throughout an image sequence. Each test evaluates an algorithm from the beginning of an image sequence, with the initialization of the appropriate ground-truth object state, until the end of the image sequence. The TRE score is the average of the tracking results of all the tests. TRE addresses this issue, unlike OPE, where the earlier part of a sequence is more significant because the findings from the frames after one tracking failure could be more informative.

Spatial Robustness Evaluation (SRE)

Although accurate initialization of a target object is vital for tracking algorithms, it is difficult in practice due to detector mistakes or manual labelling. OTB produces object states by slightly altering or resizing the ground-truth bounding box of a target object to see if a tracking technique is susceptible to initialization issues. It employs eight spatial shifts (four center shifts and four corner shifts) and four scale variations. The shift amount is 10% of the target size, and the scale ratio changes between 80 and 120 percent of the ground truth at 10% increments. The average of these 12 evaluations calculates the SRE score.

3.3 TempleColor Dataset

Even though color information is a valuable discriminative cue for visual inference, most modern trackers are limited to grayscale. Despite recent efforts to incorporate color into tracking, a thorough appreciation of color information's role remains lacking. The TC128 dataset takes a systematic approach to this challenge, looking at it from an algorithm and a

benchmarking perspective. On the algorithmic side, it encodes ten color models into 16 state-of-the-art visual trackers comprehensively. TC128 builds a huge set of 128 color sequences with ground truth and challenge factor annotations as a benchmark (e.g., occlusion). All the color-encoded trackers are run for a complete examination. An RGBD tracking benchmark is used for further validation. The findings demonstrate the value of encoding color information for tracking. A thorough analysis of multiple issues is carried out, including the behavior of various color models and visual tracker combinations, the degree of difficulty of each tracking sequence, and how different challenge factors affect tracking performance. The research serves as a guide, incentive, and standard for future research on color encoding in visual tracking.

As a critical topic in computer vision, visual tracking has many applications, including human-computer interaction, video surveillance, vehicle navigation, robotics, and more. Illumination changes, occlusion, position changes, rapid motion, and background clutter are familiar challenges for tracking algorithms in practice. Hence efforts are made to extract vital visual clues like shape and appearance to distinguish a tracking target from its surroundings. On the other hand, most current trackers depend solely on the grayscale version of an input sequence, ignoring the chromatic data. The possible causes for this are: (1) environmental factors such as changes in illumination can distort color information, (2) encoding color can raise the processing cost, and (3) grayscale images are sometimes adequate to achieve reasonably excellent results. TC128 demonstrates that color information improves visual tracking and that the benefit is universal across different tracking algorithms.

Several visual tracking algorithms have encoded color information to capture chromatic information, including newer ones that attain state-of-the-art performance. Despite these efforts, systematic research on the effects of utilizing color for visual tracking is limited. TC128 is the first complete study of color encoding in visual tracking, and it examines the problem from two perspectives: algorithm and benchmark. TC128 builds a set of 160 trackers by integrating alternative color representations and existing visual trackers, as inspired by previous work on color descriptor evaluation. Specifically, ten different color models cover various chromatic features, and 16 state-of-the-art grayscale visual trackers obtain top results in recent tracking assessments.TC128 builds a set of 128 color sequences with ground truth annotation to handle the lack of acceptable datasets on the benchmark side. 78 of the 128 sequences have never been utilized for visual tracking before, and they are often more complex than those examined previously. Each sequence's challenge variables (occlusion and out-of-plane rotation) are also included, allowing for a complete performance review.

Fig. 3.3 Initial frames of sample sequences from the TempleColor dataset with the target object depicted in a rectangular bounding box.

3.3.1 Color Tracking Benchmark

TColor-128, a color tracking benchmark, is built as an extensive dataset containing 128 color sequences to address the issue of color-based visual tracking. Previous studies and new collections are the primary sources of sequences in TColor-128. The first part consists of 50 commonly tested color sequences from previous studies. These sequences, however, need to be more sufficient for analysing color trackers thoroughly. On the one hand, because there are so many variables in visual tracking and so many tuneable parameters in visual trackers, more than 50 sequences may be needed to establish a meaningful conclusion. On the other hand, due to their popularity in prior research, performance on these sequences is frequently

near saturation, and many of them are easier than they appear. The two observations motivate the second section of TColor-128, consisting of different color sequences.

TColor-128's second section offers 78 additional color sequences gathered from the Internet. The 78 sequences, by design, are more diverse and challenging than the first 50: they take place in various settings, including a highway, an airport terminal, a railway station, a concert, and so on; none of them explicitly taped to test visual tracking algorithms; and they include a variety of challenge factors, including complete target occlusion, large illumination changes, significant target deformation, and low resolution. Figure 3.3 shows sample sequences from the TempleColor dataset.

Each sequence in TColor-128 has annotations for its challenge factors and ground truth. Scale variation (SV), illumination variation (IV), motion blur (MB), occlusion (OCC), out-of-plane rotation (OPR), deformation (DEF), fast motion (FM), in-plane rotation (IPR), out-of-view (OV), background clutters (BC), and low resolution (LR) are among the 11 factors used in TColor-128. Scale variation occurs when the bounding box size of the current frame is greater than the size of the bounding box in the first frame, with the respective ratio falling within the range $[0.5, 2]$. Fast motion occurs when the target motion is greater than 20 pixels, and low resolution is determined when the number of pixels in the ground truth bounding box is less than 400 pixels.

3.3.2 Evaluation Metrics

Evaluation of the tracking algorithms uses two commonly known metrics: the Area Under Curve (AUC) and Overlap ratio (S). The Area Under Curve (AUC) is a measure obtained from the success plot of tracking algorithms. The overlap ratio (S) is the primary measure for tracking success for each frame, given the tracking output bounding box (r_t) and ground truth bounding box (r_g), and is computed as: $S = \frac{r_t \cap r_g}{r_t \cup r_g}$ where $|.|$ denotes the area. The tracker's success rate on a sequence is the percentage of frames whose overlap score S exceeds a certain threshold. The threshold is adjusted from 0 to 1 to generate the success plot and compute the AUC. The precision plot is another parameter used in the analysis, based on the center location error (CLE), computed as the Euclidean distance between the estimated target center and the ground truth target center. The average CLE over all frames has traditionally been used to assess tracking performance. However, when a tracker loses track of the target, such measurements can become meaningless. The precision plot tackles this issue by plotting the precision versus the CLE threshold, defined as the fraction of frames with CLEs less than a threshold. The performance is ranked using the precision at the threshold of 20. One-pass evaluation (OPE) is employed to evaluate the performance of different state-of-the-art trackers extensively.

3.4 Unmanned Aerial Vehicle Dataset

UAV123 proposes a new aerial video dataset and benchmark for low-altitude UAV target tracking and a photo-realistic UAV simulator to use in conjunction with tracking methods. On 123 fully annotated and unique HD video sequences acquired from a low-altitude aerial perspective, the UAV123 benchmark presents the first evaluation of various state-of-the-art and popular trackers. The most suited trackers for UAV tracking have the highest tracking accuracy and lowest run-time among the compared trackers. The simulator can test tracking algorithms in real-time settings before deploying them on a UAV and create synthetic, photo-realistic tracking datasets with automatic ground truth annotations to augment existing real-world datasets. The benchmark and simulator have been made freely available to the vision community to encourage more study into object tracking from UAVs.

Despite decades of work on this crucial topic, visual tracking is still challenging to solve. Testing visual tracking algorithms on well-known video benchmarks, including OTB50 [62], OTB100 [70], VOT2014 [71], VOT2015 [72], TC128 [64], and ALOV300++ [73], is a widely used evaluation method. The annotated dataset must include a comprehensive selection of real-world scenarios and a distribution of tracking challenges. The benchmark is also helpful in determining future research paths in the field and how to create more robust algorithms. These well-established benchmarks now require a complete set of annotated aerial datasets that address the many issues posed by unmanned aircraft flights.

Augmenting unmanned aerial vehicles (UAVs) with automated computer vision capabilities (e.g., tracking, object/activity detection, etc.) is becoming a primary research focus, thanks to the growing availability of low cost, commercially available UAVs. Indeed, aerial tracking has enabled numerous new applications in computer vision beyond surveillance, including search and rescue, wildlife monitoring, crowd monitoring, navigation, obstacle avoidance, and sports videography. Aerial tracking can track a wide range of objects (people, animals, cars, boats, and so on), many of which are physically or persistently impossible to follow from the ground. Real world aerial tracking scenarios, in particular, present fresh challenges to the tracking problem, revealing new research fields.

This benchmark adds to existing benchmarks by establishing the aerial tracking component and providing a more extensive sampling of tracking nuisances common in low-altitude UAV videos. UAV123 is the first benchmark to examine and compare the performance of cutting-edge trackers on a large sample of annotated aerial sequences with known tracking issues. As UAV technology progresses and target trackers improve, this dataset and associated tracker evaluation offer a baseline to be helpful in the future. Visual tracking on UAVs is a potential application because the camera follows the target object based on visual feedback and actively modifies its orientation and location to maximize tracking

efficiency. Because the present benchmarks are pre-recorded sequences, they cannot provide a quantitative estimate of how slower trackers will impair the UAV's shadowing performance. UAV123 suggests using a photo-realistic simulator to render real-world landscapes and a variety of life-like moving targets present in unmanned aerial video. The simulator can test any tracker (written in Matlab or C++, for example) in various photo-realistic circumstances.

Fig. 3.4 Initial frames of sample sequences from the UAV123 dataset with the target object depicted in a rectangular bounding box.

3.4.1 Dataset

Low-altitude UAV videos differ fundamentally from videos taken in popular tracking datasets such as OTB50, OTB100, VOT2014, VOT2015, TC128, and ALOV300++. As a

result, a new dataset, UAV123, features sequences from an aerial viewpoint, with a subset intended for long-term aerial monitoring (UAV20L). Compared to the initial frame, the bounding box size and aspect ratio fluctuation are substantially more significant in UAV123. In addition, because the camera is on the UAV, it may move with the target, resulting in longer tracking sequences on average.

After ALOV300++, the UAV123 dataset has 123 video sequences and more than 110K frames, making it the second-largest object tracking dataset. Since OTB50 is a subset of both OTB100 and TC128, the total number of unique frames in all three datasets is around 90,000. Both the VOT2014 and VOT2015 datasets are subsets of existing datasets. As a result, while the tracking community has access to various datasets, the number of different sequences is lower than predicted, and sequences particular to tracking from a UAV vantage point are few.

3.4.2 Attributes

UAV123 has a wide range of scenes (including urban landscapes, highways, buildings, fields, beaches, and a harbor/marina), targets (including automobiles, trucks, boats, people, groups, and aerial vehicles), and actions (including flying) (e.g., walking, driving, cycling, flying, swimming, and wakeboarding). Occlusion, camera motion, scale variation, illumination variation, viewpoint change, background clutter, and other visual tracking challenges are all present in these sequences. Figure 3.4 shows sample sequences from the UAV benchmark dataset.

3.4.3 Evaluation Methodology

Precision and success scores compare trackers, as per the OTB50 evaluation technique. The distance between the centers of the estimated bounding box and the associated ground truth bounding box calculates the precision. The precision plot displays the percentage of tracker bounding boxes within a certain threshold distance in ground truth pixels. The trackers are ranked using a traditional threshold of 20 pixels. The intersection of pixels in the tracker bounding box and those in the ground truth bounding box determines success. The percentage of tracker bounding boxes with an overlap score exceeding a particular threshold is shown in the success plot. Another metric to rank the trackers is the area under the curve (AUC). A spatial robustness evaluation (SRE) is conducted in addition to the one-pass evaluation (OPE). The initial bounding box is scaled by 80, 90, 110, and 120 percent and spatially shifted by four center shifts and four corner shifts for SRE.

3.5 Chapter Summary

There are several single object tracking datasets available for state-of-the-art comparison and research. This chapter describes the four popular benchmark datasets- VOT, OTB, UAV, and TempleColor, used for experimental analysis in the proposed works. Details on the nature of datasets, the respective evaluation toolkits, and methodologies used to evaluate a tracking algorithm's performance are provided. The evaluation metrics used to compare and rank the trackers in respective benchmarks are also discussed. These datasets include a variety of object classes and are labelled with all the major challenges faced in the single object tracking scenario. Hence, they provide a good evaluation of the proposed methods under various challenging conditions and the overall tracking performances.

Chapter 4

Illumination and rotation adaptive correlation filters

An object tracking algorithm learns the object appearance from a single bounding box in the first frame and estimates output bounding boxes that describe the object location in all subsequent frames until the last frame. Object tracking is challenging because the object, environment, and camera used to capture the video behave inconsistently in all frames, drastically changing the appearance of the object and the background throughout the video. As a result, a tracking algorithm must deal with a wide range of scenarios in a single video. A tracking algorithm, for example, may be good at handling object scale changes but fails to locate the target due to changes in object orientations that completely modify the target appearance learned by the tracker from the first frame. An efficient tracking algorithm should track the target indefinitely, regardless of appearance or background changes, without compromising its performance.

All tracking algorithms use the same basic approach of estimating similarity across several video frames to identify the target. The object appearance model is trained both online and offline. A pre-trained network, trained on a wide range of object classes, forms the offline appearance model. These models fail to track a new class of objects on which they have not been trained. Online appearance models update themselves throughout the tracking process to address this issue, learning the object's appearance after a pre-determined number of frames to cope with the object's changing appearance.

Although several deep trackers have demonstrated their competence in various challenging sequences, their slow learning ability, and the lack of supervised data in tracking prove to be a hurdle to online learning of target appearance models. The computational efficiency and tracking accuracy of the correlation filter-based trackers have made them quite popular. They learn the object's appearance in real time from each processed frame of a video

sequence and then cross-correlate the search image from the current frame with the learned model to predict the target location in the current frame. Due to their tracking accuracy and robustness, various correlation trackers have gained widespread acceptance in the tracking community. Diverse consistency techniques have been incorporated into the basic correlation trackers, such as scale adaptiveness, windowing, bounding box regression, and so on, and have produced appealing results on various sequences.

Despite many developments in the object tracking paradigm, there is still a need to detect oriented bounding boxes based on the target object's orientation. Compared to its initial appearance, changes in object orientation can produce substantial changes in the object's appearance. These changes limit the number of robust training features extracted from each video frame, resulting in the tracker failing to detect the orientated object due to differences in the learned appearance model. Oriented bounding boxes will significantly improve the overlap ratio with the ground truth bounding box and estimate the trajectory more efficiently. Oriented bounding boxes can also aid in improving the model update and extracting better features from the rotated samples used to train the model. This work focuses on determining the object orientation and generating oriented bounding boxes that will better update the model.

The following are the main contributions of the proposed method:(a) Extraction of oriented features from each video frame helps bring rotation and deformation invariance in tracking, (b) Formulation of an object localization function eliminates false positives during detection, and (c) A position update technique that considers both the distance and direction of motion reduces inconsistencies in the estimated object track. Furthermore, we have integrated the contributions with two correlation-based trackers, Spatially Regularized Discriminative Correlation Filters (SRDCF) [2] and Efficient Convolutional Operators for tracking (ECO) [10], to demonstrate that they are generic. These trackers are chosen as the base because of their superior performance and high working speeds. These contributions are generic to be integrated into any correlation filter tracker to improve the tracking ability. In Figure 4.1, the overall architecture of the proposed method is shown.

4.1 Related Works

A variety of CF trackers have been proposed by adding restrictions to the fundamental filter design and using different feature representations of the target object. The KCF [61] tracker is the first extension, and it leverages a kernel method to conduct efficient computations in the Fourier domain. The Structural CF tracker [48] employs a part-based approach to track each object component separately utilizing different CFs. The SRDCF tracker [2] uses a

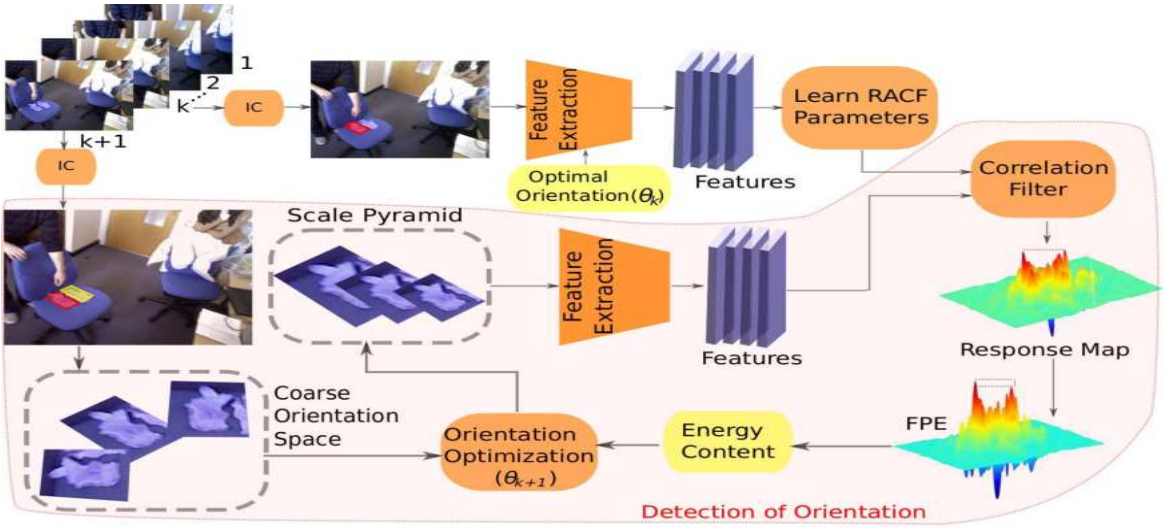


Fig. 4.1 As the pipeline indicates, both train (k^{th}) and test $(k+1)^{th}$ frames undergo illumination correction (IC) prior to feature extraction. The training features are then used to learn the parameters of Rotation Adaptive Correlation Filter (RACF). During detection stage, each candidate patch passes through a coarse orientation space from which the orientation optimizer picks a seed orientation. The seed orientation is usually the object's immediate previous orientation which is then used by Newton's iterative optimization scheme as initial point to determine optimal orientation for $(k+1)^{th}$ frame. The optimizer maximizes the total energy content in the False Positive Eliminated (FPE) convolutional response map. The response map corresponds to the winning scale in the scale pyramid. Note that the optimal orientation in the first frame (θ_1) is assumed to be 0° without loss of generality. Thereafter, the optimal orientations in the subsequent frames are determined through a deterministic optimization strategy.

spatial regularizer to weigh the CF coefficients and emphasize target areas while suppressing background information. As a result, the SRDCF tracker includes more negative patches in its training, resulting in a more discriminative model.

Earlier trackers directly represented the target object using image intensities. Color transformations [61, 74–76], Colornames [11], and other feature representations were later employed in CF trackers. Deep neural networks have progressed significantly in object detection, and deep features have found applications in tracking, resulting in considerable performance gains. Deep trackers like DeepSRDCF [34], MDNet [77], and TCNN [78] demonstrate the ability of deep networks in feature extraction. The HCF tracker [28] uses a pre-trained CNN to learn semantic and fine-grained information. To find the target object, it employs a multi-level correlation map. The CCOT tracker [36] employs DeepSRDCF [34] as a baseline and learns the filter in a continuous domain with multi-resolution feature

maps using an interpolation technique. The ECO tracker [10] uses a factorized convolution operator to reduce dimensionality and lower the computing costs. Instead of updating after each frame, ECO refreshes the features and filters after a predetermined no. of frames. This update at regular intervals removes redundancy and over-fitting to recently observed samples. As a result, the deep feature-based ECO tracker performs admirably on various datasets, surpassing other CF trackers.

According to [26], an exhaustive template search in joint scale and spatial space locates the target position and then learns a rotation template by changing the training samples to the LogPolar domain. Unlike the exhaustive search, we train a rotation-adaptive filter in the cartesian domain by integrating orientation into the conventional DCF. The proposed method embeds rotation adaptiveness directly in the basic DCF formulation, in contrast to a recent rotation adaptive method in [79], by performing a pseudo optimization on a coarse grid in the orientation space, leading to robust CF training.

The tracker in [80] employs a multi-oriented Circulant Structure with Kernel (CSK) tracker to obtain various translation models. The KCF tracker is the foundation of any translation model. The model with the best response estimates the object's location. The primary distinction is that, unlike multi-oriented CSK, the proposed tracker does not learn several translation models at various orientations to save computational costs. In contrast, we maximize the overall energy content in convolution responses with the object's orientation at the detector step. Then, a deterministic method learns the multi-channel correlation filter using a set of appropriately oriented training samples.

4.2 Illumination Correction (IC) Filter

Waving tree branches, low contrast patches, shadows of other objects, and other dynamically changing environmental factors cause illumination changes in a video. Low-frequency interference, one of the most common causes of object appearance change, is caused by this fluctuating illumination. The learned model fails to detect an object when its appearance changes substantially under different lighting conditions, resulting in a loss of accuracy and robustness. Also, at times, high-frequency changes, such as edges, which are part of the dominating features in depicting an object, may be of interest. Despite the extensive research on these difficulties, even state-of-the-art trackers need to pay more attention to them.

Though deep features have been relatively insensitive to random fluctuations in input images, such as blur, white noise, and illumination variation, the experimental results in Section 4.6 reveal that deep trackers also fail to track the object when faced with similar problems. As a result, we include an Illumination Correction filter (IC) in the tracking

paradigm to address the concerns mentioned above to some extent while retaining the regular tracked instances. To change the intensities of each frame, we first use a typical contrast stretching technique [81]. The contrast-stretched image is subjected to unsharp masking [81], a popular image-enhancing strategy, to minimize low-frequency interference and increase high variations. Surprisingly, simple enhancement of the input images significantly improves the performance of baseline trackers, as shown in Section 4.6. These improvements confirm that robust feature extractors still lack high-quality visual inputs, which may otherwise lead to significant performance gains.

4.3 Learning Rotation Adaptive Correlation Filters

In light of standard SRDCF [2], this section describes the training and detection phases of rotation adaptive correlation filters. The proposed method utilizes the same notations as SRDCF for the convenience of comprehension and to clearly distinguish the contributions. The usual SRDCF training and detection procedure is discussed first, and then, unlike heuristic template search [26, 79], rotation adaptiveness with false-positive elimination is included in the CF optimization framework.

4.3.1 Spatially Regularized Discriminative Correlation Filters

The basic DCF formulation learns a multi-channel correlation filter from a set of training samples $\{(x_k, y_k)\}_{k=1}^{t}$. Each training sample x_k is represented by a d-dimensional feature map extracted from the search region and is of resolution $M \times N$. x_k^l denotes each channel, $l\varepsilon\{1,2,...d\}$, of the feature map x_k. The response map, y_k, of each training sample x_k is a scalar valued function. A multi-channel correlation filter f with d layers is learned through the training process, where each layer is a $M \times N$ convolution filter f^l. The learned correlation filter f is applied on a test sample z and the resulting response map is computed as:

$$S_f(z) = \sum_{l=1}^{d} z^l * f^l \tag{4.1}$$

where $*$ denotes circular convolution. The correlation filter f is obtained by minimizing the L^2-error between correlation response $S_f(x_k)$ of training sample x_k and the actual response y_k as:

$$\varepsilon(f) = \sum_{k=1}^{t} \alpha_k ||S_f(x_k) - y_k||^2 + \sum_{l=1}^{d} ||\frac{w}{MN}.f^l||^2 \tag{4.2}$$

where . denotes element-wise multiplication. The correlation filter f is efficiently computed in the Fourier domain by minimizing Eq. 4.2 w.r.t. DFT coefficients.

$$\hat{\varepsilon}(\hat{f}) = \sum_{k=1}^{t} \alpha_k ||\hat{x}_k^l . \hat{f}^l - \hat{y}^k||^2 + \sum_{l=1}^{d} ||\frac{\hat{w}}{MN} . \hat{f}^l||^2 \tag{4.3}$$

Here,ˆdenotes the DFT of a function. The learned DFT coefficients $\hat{f}$ of filter f is applied on all cyclic shifts of a test sample z in a sliding-window fashion. Let $\hat{s} := \mathscr{F}\left\{S_f(z)\right\} = \sum_{l=1}^{d} \hat{z}^l \cdot \hat{f}^l$ represent the DFT of the convolution response $S_f(z)$ evaluated on the test sample z. The convolution response $s(u,v)$ at continuous location $(u,v) \in [0,M) \times [0,N)$ is interpolated by,

$$s(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \hat{s}(m,n)\, e^{i2\pi\left(\frac{m}{M}u + \frac{n}{N}v\right)}. \tag{4.4}$$

where i denotes the imaginary unit. The maximal sub-grid location (u^*, v^*) is then obtained using Newton's method, by optimizing $argmax_{(u,v)\in[0,M)\times[0,N)} s(u,v)$, starting at maximal grid-level score $(u^{(0)}, v^{(0)}) \in \Omega$. Thus, the standard SRDCF efficiently incorporates translation invariance by exploiting the periodic assumption with spatial regularization, but this does not implicitly learn rotation adaptiveness. Therefore, the proposed method extends the discriminative capability of SRDCF by learning rotation adaptive correlation filters through a deterministic optimization approach.

4.3.2 Orientation Adjusted Discriminative Correlation Filters

Like the use of oriented sample images in the similarity-based deep trackers, orientation correction in correlation trackers can be done by learning the correlation filters from oriented training samples. This section goes over the filter optimization technique in SRDCF in great depth. This method may be used for any correlation tracker, and we demonstrate the technique's applicability by incorporating it into the ECO tracker.

From correctly oriented training samples, we learn orientation adjusted spatially regularised correlation filters and solve the filter optimization problem in the Fourier domain, as in basic SRDCF. Let the orientation of training sample x_k be θ_k. It is assumed that $\theta_k = 0$ for $k = 1$. i.e., the orientation of the initial sample is assumed to be zero.

Rotating x_k anti-clockwise with an angle θ_k and cropping to the same size $M \times N$ as x_k yields the rotated training sample x_k^θ. We learn the multi-channel correlation filter in the proposed technique from appropriately oriented training samples $\left\{\left(x_k^\theta, y_k\right)\right\}_{k=1}^{t}$. The final response map on rotated training samples x_k^θ is given by:

$$S_f(x_k^{\theta}) = \sum_{l=1}^{d} x_k^{\theta l} * f^l. \tag{4.5}$$

The objective function to learn orientation adjusted correlation filters becomes:

$$\varepsilon_{\theta}(f) = \sum_{k=1}^{t} \alpha_k \left\| S_f(x_k^{\theta}) - y_k \right\|^2 + \sum_{l=1}^{d} \left\| \frac{w}{MN} \cdot f^l \right\|^2. \tag{4.6}$$

The orientation adjusted correlation filters can be learned by optimizing Equation 4.6 in Fourier domain using Gauss-Seidel iterative optimization as in the base SRDCF [2].

$$\hat{\varepsilon}_{\theta}(\hat{f}) = \sum_{k=1}^{t} \alpha_k \left\| \sum_{l=1}^{d} \hat{x}_k^{\theta l} \cdot \hat{f}^l - \hat{y}_k \right\|^2 + \sum_{l=1}^{d} \left\| \frac{\hat{w}}{MN} * \hat{f}^l \right\|^2. \tag{4.7}$$

The proposed method requires appropriately oriented training samples to carry out the training procedure. The response score and orientation of a test sample z extracted from the $(t+1)^{th}$ frame are computed using the correlation filter f learned from training samples of the t^{th} frame. Thus, in the $(t+1)^{th}$, i.e., 2^{nd} frame, we learn the filter coefficients of f from $(x_{k=1}^{\theta=0^{\circ}}, y_{k=1})$ and identify the object location, (u_{k+1}^*, v_{k+1}^*), and orientation, θ_{k+1}.

We evaluate the learned filter on oriented test samples at the detection stage to get the response maps for oriented selections. Let $\hat{s}_{\theta} = \mathscr{F}\left\{S_f(z^{\theta})\right\} = \sum_{l=1}^{d} \hat{z}^{\theta_l} \cdot \hat{f}^l$ represents the DFT ($\mathscr{F}\{.\}$) of convolution response $S_f(z^{\theta})$, evaluated at θ orientation of test sample z. Similar to Equation 4.4, we compute $s_{\theta}(u,v)$ on a coarse grid $(u,v) \in \Omega$ by,

$$s_{\theta}(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \hat{s}_{\theta}(m,n)\, e^{i2\pi\left(\frac{m}{M}u + \frac{n}{N}v\right)}. \tag{4.8}$$

Here, orientation space $\Phi = \{\theta_k \pm a\delta\}$, where $a = 0, 1, 2, \ldots, A$. As a result, the orientation space Φ consists of $(2A+1)$ rotations with δ step size. We utilized $\delta = 5^{\circ}$ and $A = 2$ in our studies since an object's orientation is less likely to alter dramatically across consecutive frames. The false-positive elimination technique computes the target centroid and final orientation.

In any correlation tracker, oriented training samples can be utilized similarly to SRDCF to learn orientation corrected filters, with the only difference being the cost function to be optimized. The results section provides the results of applying the orientation adjustment to ECO.

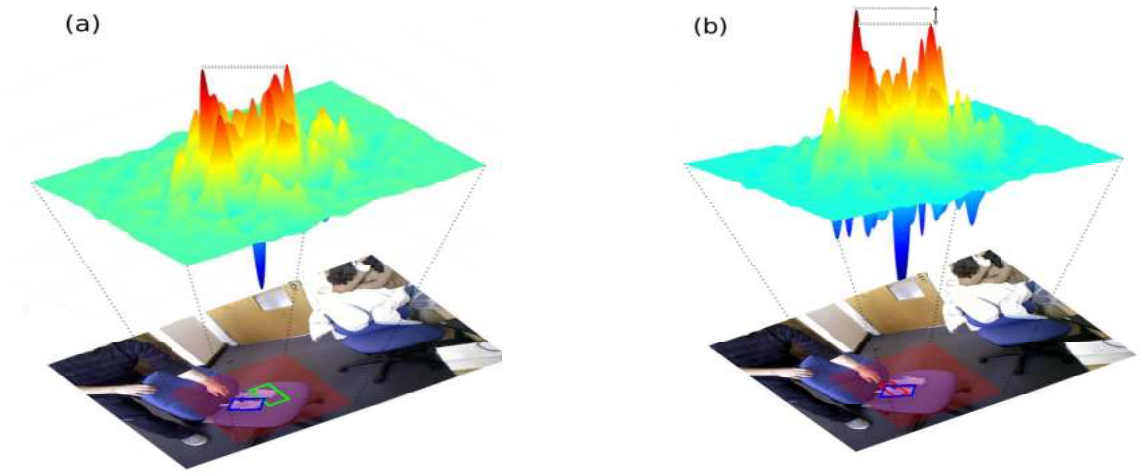


Fig. 4.2 Sample frames from the sequence glove of VOT2016 [1]. The blue, green, and red rectangle shows the output of groundtruth, ECO, and F-ECO (with FPE), respectively. Convolution response of shaded (red) region obtained directly (a) without, and (b) with optimization through false positive elimination.

4.4 False Positive Elimination (FPE) and Target Estimation

It was observed that the response map of test samples could sometimes contain numerous peaks with similar detection scores after conducting extensive trials. This often happens when test samples are collected from an image region with multiple objects having comparable representations to the target object. As a result, rather than directly selecting the top response location, we propose using a new metric called Score to Distance Ratio (SDR), defined as the ratio of response at each location to the distance from the previous target center.

A separate SDR map is computed for each response map generated. The target scale, orientation, and centroid in the current frame are used to estimate the scale, orientation, and position of the maximum SDR score. We hope to find an object with a high response score and the slightest deviation from its original location. This formulation claims that an object is less likely to deviate significantly from its previous location.

Consider the glove sequence from the VOT2016 dataset [1]. Figure 4.2(a) shows that both gloves have the same response score due to comparable feature representation. As illustrated in Figure 4.2(b), the SDR score mitigates this problem by maximizing the response score while keeping the divergence from the previous centroid to a minimum. Unlike SRDCF which maximizes $s(u,v)$ alone, we propose to maximize the SDR computed as $\frac{s(u,v)}{\left\|\left(u-u_k^*, v-v_k^*\right)\right\|}$.

Here, (u_k^*, v_k^*) denote the sub-grid level target location in the k^{th} frame. Again, the new target location corresponds to the one with the highest SDR score.

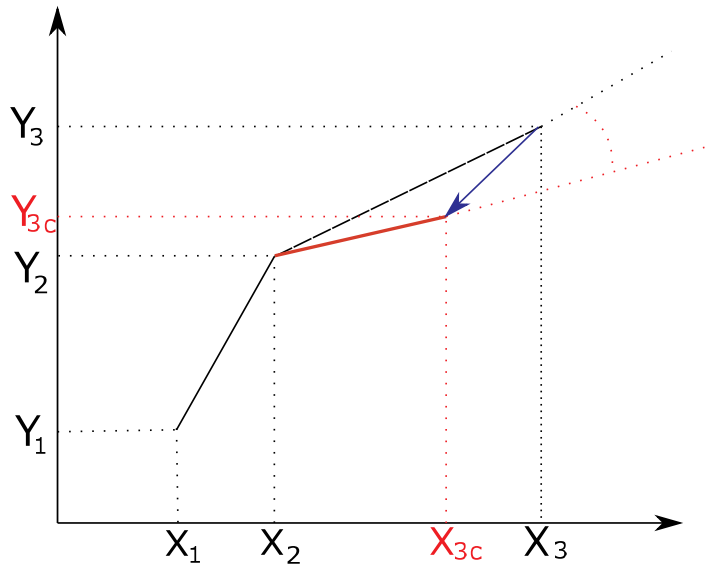


Fig. 4.3 Conventional centroid update technique. Let $[X_1, Y_1]$ and $[X_2, Y_2]$ represent the centroids in the first and second frames respectively. Let $[X_3, Y_3]$ represent the predicted centroid in the third frame. Let $[X_{3c}, Y_{3c}]$ represent the updated centroid in the third frame. Let δ represent the angular deviation due to conventional centroid update.

4.5 Displacement Consistency

Traditional tracking algorithms may deviate from the original target path due to intrinsic restrictions, and the entire track may be lost. We may prevent this drift by checking the distance travelled by the tracker from the previous frame, which is predicated on the notion that an object does not move far from its last location. Some existing algorithms, such as [43, 46], use a position update technique, as shown in Figure 4.3, which may be mathematically expressed as Equation 4.9, to enforce smoothness on object motion.

$$[X_{3c}, Y_{3c}] = w \times [X_2, Y_2] + (1 - w) \times [X_3, Y_3] \tag{4.9}$$

The incremental angular deviation δ, on the other hand, leads the target centroid to deviate from the actual centroid. It was observed that smoothness in the object's displacement contributes more to accurate tracking when both the distance and direction of motion are considered. As a result, a new displacement consistency technique is proposed, that incorporates both distance and angular consistency to improve smoothness of object motion.

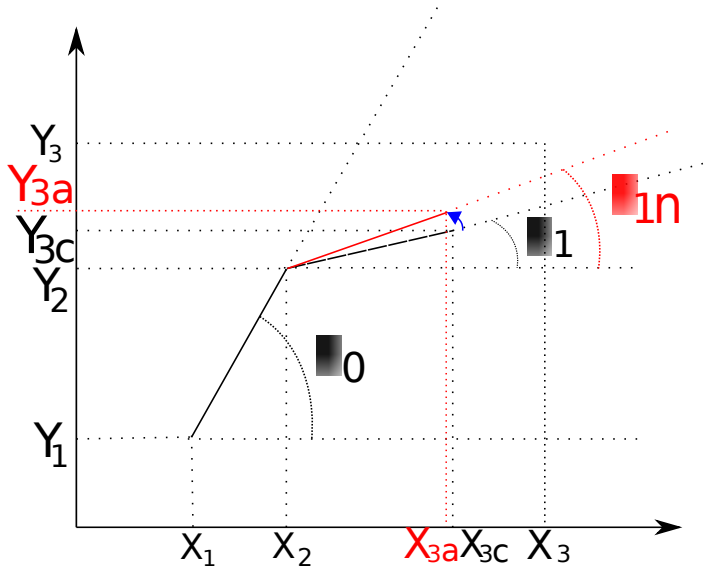


Fig. 4.4 Angle consistency. Let θ_1 represents the angle of the centroid in the third frame with respect to the centroid in the second frame. Let θ_0 represents the angle of the centroid in the second frame with respect to the first. Let θ_{1n} represents the updated angle in the third frame. Let $[X_{3a}, Y_{3a}]$ represents the new updated centroid by using equation (4.10) with 1% weight given to previous angle i.e. $w_\theta = 0.01$.

Figure 4.4 shows the angular consistency, calculated mathematically as a rolling average of the angle moved in the previous and present frames.

$$\theta_{1n} = w_\theta \times \theta_0 + (1 - w_\theta) \times \theta_1 \tag{4.10}$$

The total distance moved by the target is updated as a rolling average of the previous, and current distances travelled. Figure 4.5 depicts distance consistency, which is calculated mathematically as:

$$d_{1n} = w_d \times d_0 + (1 - w_d) \times d_1 \tag{4.11}$$

Using Equation 4.12, the displacement consistency technique calculates the new centroid, considering the average angle and distance travelled.

$$[X_{3n}, Y_{3n}] = [X_2, Y_2] + d_{1n} \angle \theta_{1n} \tag{4.12}$$

The CFNet trackers estimate the final target centroid using Equation 4.12.

The sub-grid location, $left(u*k+1, v*k+1right)$, given from equation(4.13), is updated in correlation filter trackers by,

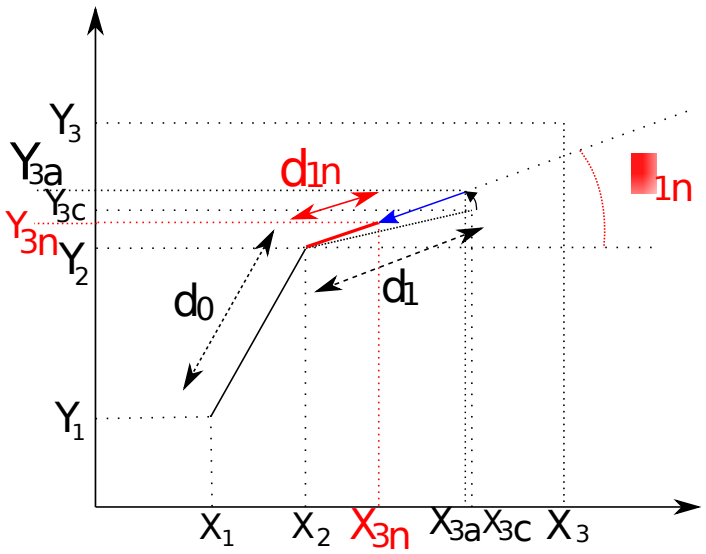


Fig. 4.5 Distance consistency. Let d_0 represents the distance of centroid from frame 1 to 2. Let d_1 represents the distance from frame 2 to 3 after angle consistency. Let d_{1n} represents the updated distance obtained by using equation (4.11) with 1% preference given to previous distance i.e. w_d = 0.01. Let $[X_{3n}, Y_{3n}]$ represents the final position of the centroid after Displacement consistency.

$$\begin{aligned}\left(u^*_{k+1}, v^*_{k+1}\right) &= \left(u^*_k, v^*_k\right) + d_{1n}\angle\varphi_{1n},\\ d_{1n} &= \omega_d \times d_1 + (1-\omega_d) \times d_0,\\ \varphi_{1n} &= \omega_a \times \varphi_1 + (1-\omega_a) \times \varphi_0,\end{aligned} \tag{4.13}$$

where, $d_0 = \left\|\left(u^*_k - u^*_{k-1}, v^*_k - v^*_{k-1}\right)\right\|$, $d_1 = \left\|\left(u^*_{k+1} - u^*_k, v^*_{k+1} - v^*_k\right)\right\|$, $\varphi_0 = \arctan\left(u^*_k - u^*_{k-1}, v^*_k - v^*_{k-1}\right)$, $\varphi_1 = \arctan\left(u^*_{k+1} - u^*_k, v^*_{k+1} - v^*_k\right)$, $\omega_d = 0.9, \omega_a = 0.9$.

By lowering the contribution of d_1 and φ_1 to 0.9, the abrupt transition from $\left(u^*_k, v^*_k\right)$ to $\left(u^*_{k+1}, v^*_{k+1}\right)$ is prevented.

4.6 Experimental Results

We describe the experimental setup first, then conduct ablation investigations to determine the impact of each component on total tracking performance. Then, using the VOT [82] and OTB100 [70] benchmarks, we undertake thorough tests to compare various state-of-the-art trackers qualitatively and quantitatively. For evaluation on these benchmarks, we used the VOT and OTB toolkits in all our tests.

4.6.1 Implementation Details

We assess all models, including baseline SRDCF and ECO, on the same system under the same experimental setup to undertake an unbiased analysis that may arise due to varying numerical precision of different systems. The tests are run on a single computer with an Intel(R) Xeon(R) CPU E3-1225 v2 @ 3.20GHz, four cores, four logical processors, and 16GB RAM. Matconvnet was used to implement the proposed tracker in MATLAB. Apart from the additional parameters $\delta = 5^\circ$ and $A = 2$ in rotation adaptive filters, we utilize the same parameter values as the baseline. These options were chosen since the orientation does not change significantly between frames. In IC, we use a contrast stretching output intensity range of $[0, 255]$ and an unsharp masking threshold of 0.5. This intensity range was chosen to correspond to the most common image representation (uint8). This threshold, on the other hand, is manually set based on qualitative results.

4.6.2 Ablation Studies

For faster experimentation, we gradually integrate Displacement consistency (D), False positive elimination (F), Rotation adaptiveness (R), Illumination correction (I), and their combinations into the ECO framework and assimilate the impact of each component on Average Expected Overlap (AEO), the standard metric on the VOT benchmark. During the development process, we test each ablative tracker on a collection of 16 videos (Table 1) selected from 60 videos in the VOT dataset. If annotated with challenges like deformation, rotation, or illumination change, a video frame is chosen. It is worth noting that the FPE scheme improves performance in every integration, while illumination correction alone improves RDF-ECO performance by 7.7%. The proposed ideas, both individually and collectively, provide a good improvement relative to the base model, as evident from Table 4.1.

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 4.1 Quantitative evaluation of the ablative trackers on a set of 16 challenging videos from the VOT benchmark.

4.6.3 Evaluation of CF trackers

We compare our results with a baseline technique and a few other CF trackers on challenging sequences from VOT to assess the overall performance of RIDF_SRDCF, as shown

in Figure 4.6. We also compare the Average Expected Overlap (AEO) of a few correlation filter-based trackers to assess performance objectively under various challenging conditions, as shown in Figure 4.7. In most of the individual categories, the suggested RIDF_SRDCF beats the traditional SRDCF, resulting in a 11.4% and 13.04% overall improvement in AEO and robustness, respectively. Illumination change, Size change, Motion Change, Camera motion, and Empty categories gain 56.25%, 23.53%, 38.46%, 5.26%, and 16.66%, respectively, in the category comparison, as shown in Figure 4.7. It is worth noting that the percentage improvement is calculated in comparison to the base SRDCF. According to our tests, the proposed rotation adaptive technique outperforms its competitors on the VOT benchmark.

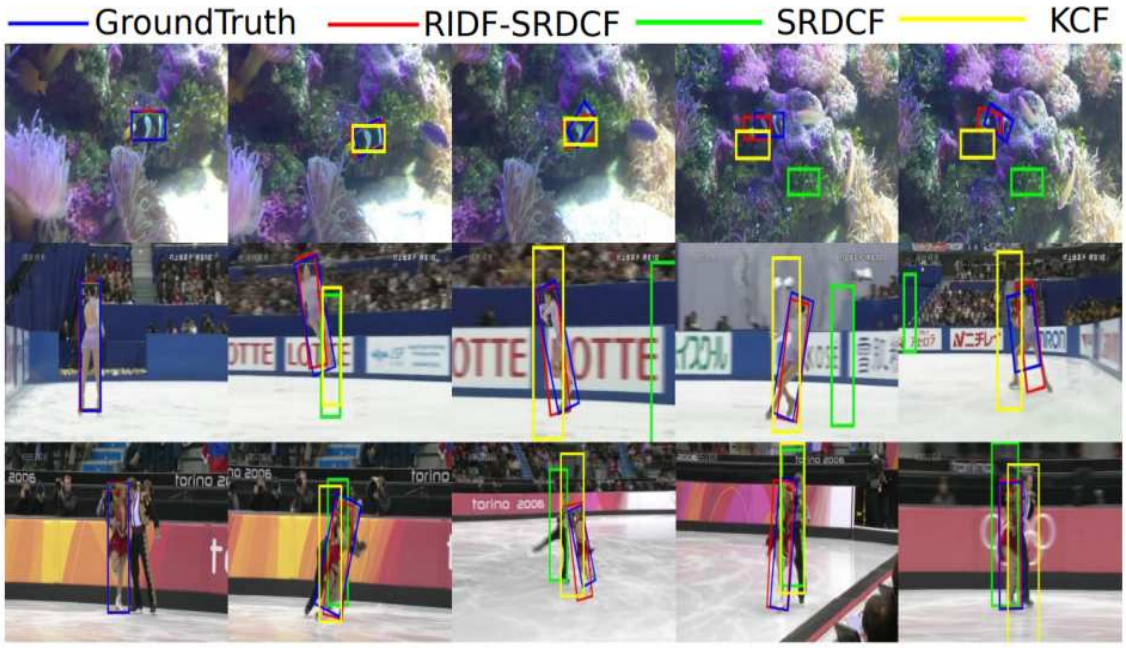


Fig. 4.6 Qualitative analysis of RIDF-SRDCF. The proposed tracker successfully tracks the target under severe rotation, unlike SRDCF and KCF. The rotation adaptive filters assist in determining the orientation of the target object effectively that leads to substantial gain in overall performance. To avoid clumsiness only few bounding boxes are plotted and other variants are quantified in Figure 4.7.

4.6.4 State-of-the-art Comparison

Extensive experimental results confirm the efficacy of proposed contributions to the holistic visual object tracking challenge. The development stage and hyper-parameter tuning employ the VOT benchmark. The proposed trackers are benchmarked using the same parameter settings on VOT and OTB to assess the generalization capability.

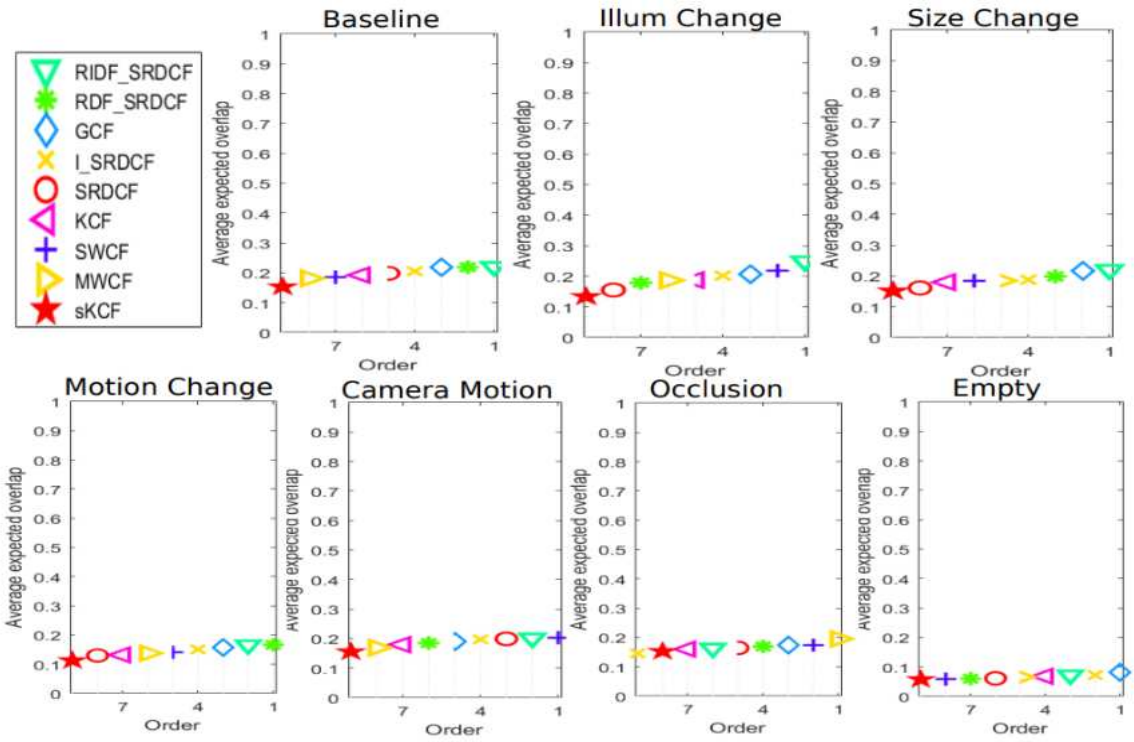


Fig. 4.7 Average Expected Overlap analysis of correlation filter based trackers

Evaluation on VOT2016

On the VOT dataset, we analyse the top-performing model from Table 4.1 and SRDCF's ablative trackers. According to Table 4.2, the I_SRDCF, RDF_SRDCF, and RIDF_SRDCF significantly improve the base SRDCF by 3.53%, 10.60%, and 11.41% in AEO and 4.83%, 17.87%, and 13.04% in robustness respectively. With a small improvement of 1.71% in AEO and as high as 6.41% in robustness, the RIDF_ECO outperforms state-of-the-art trackers such as MDNet (winner of VOT2015) and CCOT (winner of VOT2016). The percentage gain is calculated in comparison to the baseline.

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 4.2 State-of-the-art comparison on the whole VOT dataset.

Evaluation on OTB100 dataset:

In the majority of the tracking challenges, the proposed RIDF_ECO outperforms the baseline and state-of-the-art trackers, according to the OTB100 evaluation (Table 4.3). The proposed technique has problems coping with Background Clutter, Deformation, and Out-of-Plane Rotation, according to the quantitative analysis in Table 4.3. Though these modifications improve ECO's performance in various failure scenarios, it is still no match for MDNet. The SRDCF tracker with deep features, i.e., DeepSRDCF, trails behind RIDF_SRDCF in terms of success rate and precision, which are typical parameters in the OTB100 benchmark. It's worth noting that the hyperparameters utilized in the OTB100 evaluation are the same as those used in the VOT evaluation. This confirms that rotation adaptive filters and other contributions generalize effectively between the OTB100 and VOT benchmark datasets.

Trackers	RIDF_ECO	ECO	MDNet	CCOT	RIDF_SRDCF	DeepSRDCF	SRDCF	CFNet	Staple	KCF
Out of view	0.767	0.726	0.708	0.725	0.712	0.619	0.555	0.423	0.518	0.550
Occlusion	0.721	0.710	0.702	0.692	0.652	0.625	0.641	0.573	0.610	0.535
Illumination Variation	0.702	0.662	0.688	0.676	0.649	0.631	0.620	0.561	0.601	0.530
Low Resolution	0.734	0.652	0.663	0.642	0.588	0.438	0.537	0.545	0.494	0.384
Background Clutter	0.648	0.638	0.697	0.620	0.634	0.616	0.612	0.592	0.580	0.557
Deformation	0.687	0.687	0.722	0.657	0.652	0.645	0.641	0.618	0.690	0.608
In-plane Rotation	0.696	0.645	0.656	0.653	0.635	0.625	0.615	0.606	0.596	0.510
Out-of-plane Rotation	0.682	0.665	0.707	0.663	0.646	0.637	0.618	0.593	0.594	0.514
Fast Motion	0.716	0.698	0.671	0.694	0.693	0.640	0.599	0.547	0.526	0.482
Overall Success Rate	0.702	0.691	0.678	0.671	0.641	0.635	0.598	0.589	0.581	0.477
Overall Precision	0.937	0.910	0.909	0.898	0.870	0.851	0.789	0.777	0.784	0.696

Table 4.3 State-of-the-art comparison on OTB100 dataset.

4.7 Chapter Summary

We showed in this chapter that using a basic but efficient image enhancement technique before feature extraction might significantly improve the tracking paradigm. We investigated the efficacy of the proposed rotating adaptive correlation filter in a standard DCF formulation and found it to be effective on popular tracking benchmarks. The sub-grid detection approach was improved by optimizing object orientation by false-positive reduction, which improved overall performance. In addition, checking the constancy of displacement on CF monitors yielded promising results in various settings. Furthermore, because the DCF formulation is employed as the backbone of most state-of-the-art trackers, we believe that the suggested rotation adaptive method in correlation filters can be easily implemented into various frameworks and will contribute to the advancement of tracking research. The rotation adaptive correlation filters still show limited performance in scenarios like background clutter, deformation, out of plane rotations etc. To deal with the tracking challenges in general, a tracking resumption

technique needs to be integrated with the correlation filter trackers, which is discussed in Chapter 5.

Chapter 5

Detection based long term tracking in correlation filter trackers

Correlation based trackers have improved robustness to specific issues such as scale changes, rotation, illumination changes, etc. However, they still struggle to track under other conditions such as deformations, background clutter, occlusion, etc as discussed in Chapter 4. Correlation filter trackers follow an online update mechanism that learns the target appearance from each frame to adjust to changes in target appearance. As a result of an erroneously tracked frame, the filter retains the inaccurate target appearance, and the learning error increases as more frames are processed. This causes the tracker to drift away from the actual target path and continues to track the background, limiting the long term tracking capability.

The main drawback of correlation trackers for long-term tracking is that they cannot resume tracking once the track is lost or if the object moves out of the camera's field of vision. This shortcoming is due mainly to the need for a detector-like mechanism in the correlation-based tracking framework. This chapter details a tracking resumption technique to facilitate long-term tracking by incorporating a detector module into the correlation tracker framework.

The proposed detection-based long-term tracking (DLT) algorithm makes the following contributions: (i) a generic technique for incorporating tracking resumption in any correlation tracker, (ii) an adaptive threshold to decide tracking uncertainties in correlation trackers, (iii) a detector formulated in the tracking scenario using Bag-of-Visual-Words (BoVW), and (iv) tracker re-initialization scheme to eliminate model-drift. The Efficient Convolution Operators (ECO) [10] form the base correlation tracker in the proposed long-term tracking technique.

5.1 Related Works

Correlation filter trackers:

The high-speed computations of the correlation filter trackers have developed several techniques to improve their accuracy and robustness. The characteristics used to represent the target object or the cost function to create the correlation filter are the key drivers of progress in this direction. State-of-the-art trackers mostly use handcrafted features like HoG [61] or features produced from pre-trained CNNs [34]. A spatial regularisation technique is created in [2] to weigh the filter coefficients and incorporate scale handling by investigating the learning stage of the correlation trackers. [36] learns the correlation filter using an interpolation technique in the continuous domain. Using a factorized convolution operator in [10] reduces the time complexity of correlation trackers. Among CF trackers, [83] uses a kernel ridge regression technique, and the context aware tracker in [42] explicitly incorporates the global context, while [84] introduces a spatial-temporal attention mechanism. Despite all the advances, CF trackers continue to fail to track under target appearance alterations, limiting their long-term tracking capacity. The proposed method creates a detector in the tracking scenario to incorporate long-term tracking in CF trackers and re-initializes the tracker upon target loss.

Tracking by detection:

Numerous tracking-by-detection systems have become prevalent to deal with various tracking issues. TLD [60] uses the median flow tracker combined with a cascaded detector, and the three components work in tandem. Even though the TLD tracker supports tracking resumption, it fails in background clutter, object deformations, rotations, and other factors since it employs a simple Lucas–Kanade tracker via a complicated detector. [85] proposes a tracking-by-detection method based on an online random fern classifier. Existing detection-based tracking approaches necessitate the continuous application of a detection algorithm on every frame of a given sequence and a search of the entire frame for detection. Still, their performance needs to catch up to that of state-of-the-art trackers. The proposed method simulates a detector activated only on failed frames. Object detection acts on a confined search region surrounding the previous target location.

5.2 The Detection based Long-Term Tracker

This section details the key elements of the proposed tracking algorithm: tracker, detector, re-initialization, and model update.

5.2.1 Tracker component

The tracker component employs a correlation filter to locate the target object in each frame. The correlation filter tracker learns a multi-channel classifier f using a collection of training samples $\{(x_k, y_k)\}_{k=1}^{n}$, generated from the starting frame, where n is the number of training samples. The object samples x_k represents the cyclic shifts of the first target patch. A $d-$ dimensional feature map of size $M \times N$ represents each training sample. The desired correlation map y_k is a Gaussian of size $M \times N$ that shows the degree of correlation at each point of the sample x_k, with the highest value $(= 1)$ corresponding to the target center and gradually decreasing values towards the edges.

By minimizing the L^2- error between the responses on the training samples x_k and the desired correlation map y_k, a correlation filter layer corresponding to each feature layer is learned.

$$\min_{f^l} \sum_{l=1}^{d} ||x^l * f^l - y||^2 + \lambda ||f^l||^2 \tag{5.1}$$

where f^l and x^l are the corresponding filter and feature layers, $\lambda \geq 0$ is the weight of the regularisation term and $*$ represents circular convolution. In the Fourier domain, element-wise computations derive the correlation filter. The resulting correlation filters generate correlation peaks in the response map corresponding to the target center and reduced responses to the background.

The correlation filter trained in the first frame performs tracking in subsequent frames. An $M \times N$ feature map is extracted from a test sample z around the previous target location in each new frame. The learned correlation filter f is applied to the feature map to produce the correlation map $S(z)$ in the spatial domain, as follows:

$$S(z) = F^{-1}\left(\sum_{l=1}^{d} \hat{z}_l \bigodot \hat{f}_l\right) \tag{5.2}$$

where $\bigodot$ represents element-wise multiplication, $\wedge$ represents the Discrete Fourier Transform (DFT) and F^{-1} represents the inverse DFT.

5.2.2 Detector component

Many object-retrieval [86] and object-detection [87] strategies have recently gained popularity, notably deep-learning-based models. Though accurate, they confine to the (few) classes on which they are trained and rely only on the training data. Deep learning-based detection models in the tracking scenario are constrained by the computing time necessary to retrain a deep detector at frequent regular intervals and the limited training data. Therefore, providing a simple and efficient detector with quick learning capabilities in the tracking framework is critical. The proposed technique creates an object detector in the tracking scenario by learning the object appearance using BoVW. The BoVW features are parts-based representations that can accurately represent an object regardless of its transformations.

Training the Detector

The initial training of the proposed detector uses the first and second frames of the sequence. The assumption that the correlation tracker will not fail in the second frame forms the basis for generating training samples from the second frame. Scaled, rotated, and motion-blurred representations of the target in the first two frames, along with background patches from the first frame, are utilized for training a robust detector. Thus, the detector initially learns using 100 object patches (50 from each frame) and 100 background patches. The BoVW features generated using the steps below represent each training sample:

1. *Visual vocabulary creation:* The visual vocabulary consists of the SURF feature descriptors (1×64 vectors) derived from the whole set of 200 training data.

2. *Visual word computation:* The k-means clustering technique divides the visual vocabulary into k clusters. The final cluster centers are the visual words (1×64 vectors). The value of $k = 500$.

3. *BoVW representation:* A histogram of the visual words represents a training sample in the BoVW description. The visual words form the histogram bins, and the bin values correspond to the distance between descriptors from the training sample and the corresponding visual word. As a result, a k-valued feature vector represents each training sample, where k denotes the number of visual words.

An SVM learned from BoVW features of the training samples and their corresponding labels (object or background) act as the object classifier. The BoVW features form a $200 \times k$ matrix, with each row corresponding to features from a training sample, and the labels form a 200×1 vector, with each value denoting the label of the corresponding training sample.

Figure 5.1 depicts the training of the proposed detector in the tracking scenario. Upon a failure in any future frames, the object detector trained from the first and second frames is used to re-initialize the tracker.

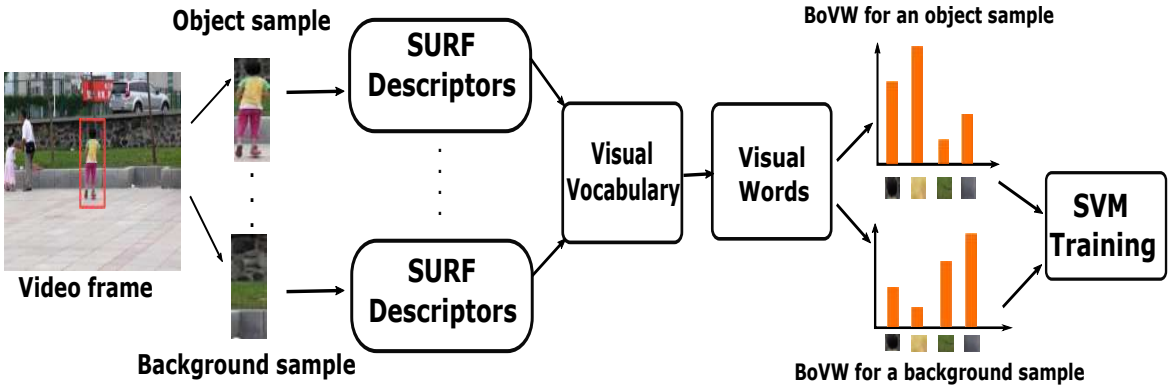


Fig. 5.1 Proposed training of the detector using BoVW representation of object and background training samples. The red box shows the target object.

Testing the Detector

Upon a target loss in a new frame, the trained detector scans a three-times-the-target-size area surrounding the previous target location to generate test patches. The detector assesses whether the object is present or absent in each patch. The scanning window creates all potential patches with the following parameters:

- Window size equals target size.
- The horizontal step equals 50% of object width.
- The vertical step equals 50% of object height.

A BoVW feature vector is created for each test sample. The trained detector will classify each test patch as either object or background. The correlation filter learned in the previous frame is applied to the classified object patches. The patch with the highest response is the detector output.

5.2.3 Tracker Resumption

Running a detector on every frame is computationally expensive. Hence the proposed method uses the detector only when the correlation-based tracker component detects a tracking failure.

5.2.4 Failure Identification

Due to numerous changes in the object or background, the CF-based tracker component loses the target track. Literature holds specific techniques for determining object occlusion or out-of-view movements. However, they do not account for other tracking errors. We present a general and adaptable metric to determine tracking uncertainty in a CF-based tracker.

The tracker's correlation response in any frame indicates the likelihood of the target appearing in that frame. As a result, a failure detection threshold based on the tracker response is appropriate. A constant threshold on the response is not meaningful in all instances. Therefore, we define an adaptive measure computed as the ratio of the tracker response in the current frame to the tracker response in the second frame. The fact that CFs learn most confidently in the first frame, where the actual target object is precisely specified, justifies this idea; hence the tracker's most dependable output is in the second frame.

A higher threshold on the response ratio forces the detector to run on a larger number of frames, increasing computations, whereas a lower threshold may miss a failure. A threshold of 0.6, chosen empirically, ensures a suitable trade-off between computational expense and failure detection. A target loss maps to a response ratio of less than 0.6 in any frame. In other words, tracking uncertainty is indicated by a tracker response of less than 60% of the tracker's response in the second frame.

A tracking failure in any frame calls the trained detector. As a result, a target loss identified in any frame generates two output bounding boxes - one from the tracker and the other from the detector.

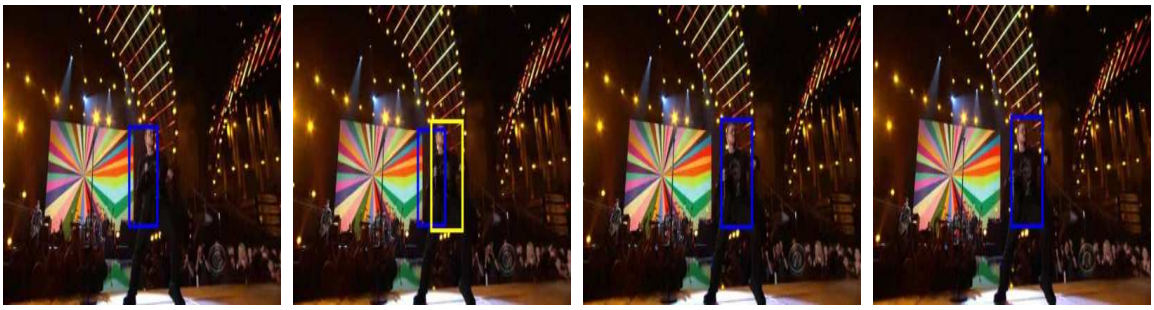


Fig. 5.2 Demonstration of tracker re-initialization using the detector mechanism. The first column shows the tracker output in frame t using the blue box. The second column shows the tracker output in frame $t+1$ using the blue box which indicates a tracking failure and the corresponding detector output is shown by the yellow box. The tracker is re-initialized using the detector output and continues to track the actual object in the remaining frames as shown in the third and fourth columns.

5.2.5 Final target estimation

A template matching technique chooses between the tracker and detector outputs. The Normalized Cross-Correlation (NCC) of two output patches with the initial target patch is determined, and the patch with the highest correlation is the final target in the current frame. The template matching technique eliminates false positives from the detector, preventing the tracker from degrading.

Figure 5.2 depicts the working of the tracker and detector on a tracking failure and the selection of the best-correlated patch as the new target to re-initialize the tracker. The blue bounding box in the first column shows the tracker output in frame t. In frame $t+1$, a tracking failure (represented by the blue bounding box) occurs when the tracker response goes below the uncertainty threshold, triggering the detector. The yellow bounding box in the second column represents the detector output. The detector patch is utilized to re-initialize the tracker and is calculated as the new target in frame $t+1$ based on the NCC of the tracker and detector outputs. As illustrated in the third and fourth columns, the tracker recovers the lost track.

5.2.6 Model Update

An online update of the tracker and detector uses training samples from each frame to adapt the tracker and detector to target appearance changes. The appearance model x and the filter f are updated frame by frame with a learning rate α as follows to make the correlation filters adaptive to target variations:

$$\hat{\mathrm{x}}^t = (1-\alpha)\hat{\mathrm{x}}^{t-1} + \alpha\hat{\mathrm{x}}^t \tag{3a}$$

$$\hat{f}^t = (1-\alpha)\hat{f}^{t-1} + \alpha\hat{f}^t \tag{3b}$$

where t is the index of the current frame. The learning rate α is selected as 0.02 as in the case of general CF trackers [61].

New training samples are added from each of the correctly tracked frames to incorporate temporal information from the video sequence and adapt the detector to the appearance changes of the object. If the tracker response in any frame exceeds the uncertainty threshold, an object patch and a random background patch extracted from the frame are added to the training set. The immediately previous ten samples are removed, and the detector is retrained for every ten new training samples added. Figure 5.3 depicts the overall workflow of the proposed DLT tracker.

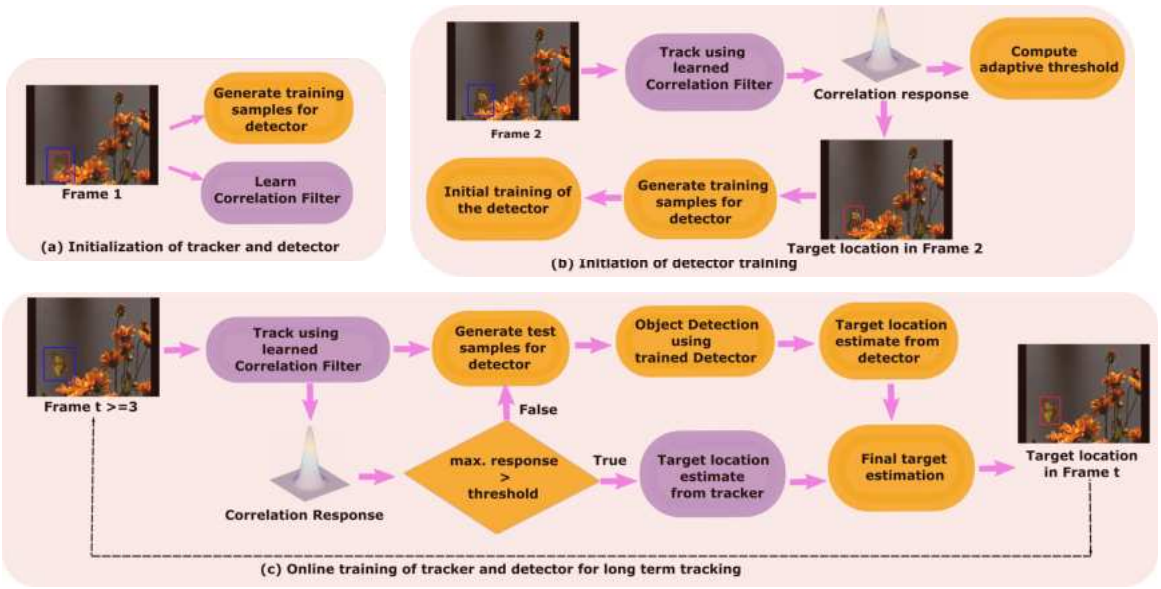


Fig. 5.3 The overall workflow of the proposed DLT tracker. The tracker components are shown in the violet boxes and detector components are shown in yellow boxes.

5.3 Experimental Results

The proposed detection-based long-term (DLT) tracking algorithm is assessed on four publicly available benchmark datasets: OTB50 [62], OTB100 [70], VOT [82], and UAV123 [63] for a generic and broad evaluation.

5.3.1 Implementation Details

All analyses, including the baseline ECO [10], were carried out on the same system with the same experimental setup. The experiments are run on a single computer with an Intel(R) Xeon(R) CPU E3-1225 v2 @ 3.20 GHz, four cores, four logical processors, and 16GB RAM. Matconvnet was used to develop the proposed technique in MATLAB.

5.3.2 Qualitative Analysis

The qualitative evaluation of the proposed algorithm on selected frames of the benchmark sequences is shown in this section. In Figure 5.4, we have demonstrated the bounding boxes of only five trackers: DLT, ECO [10], CCOT [36], Staple [14], and SiameseNet [43], where each row displays tracking outcomes under varied challenges for clarity in presentation. The proposed DLT algorithm outperforms the compared trackers, particularly the base ECO tracker, in challenging scenarios.

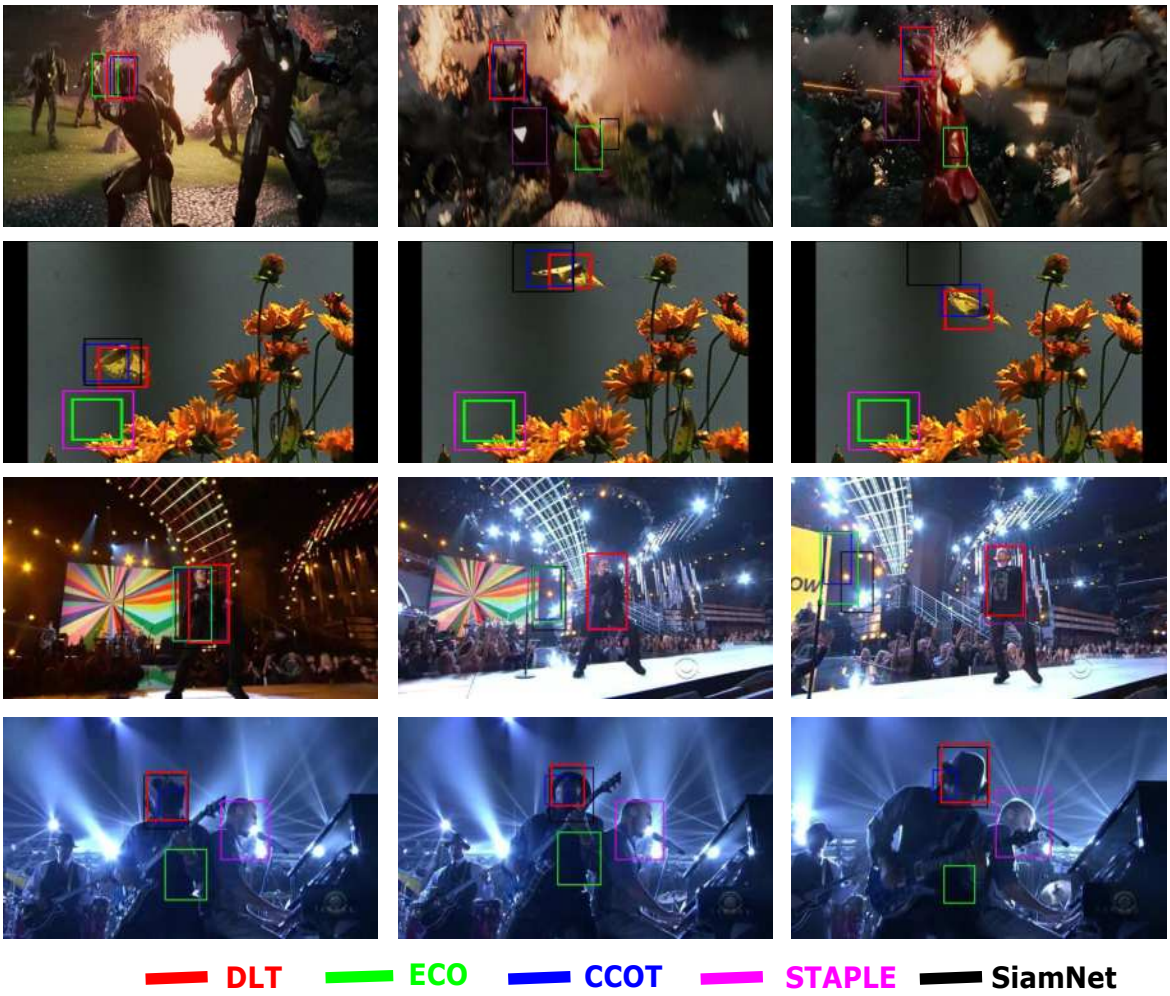
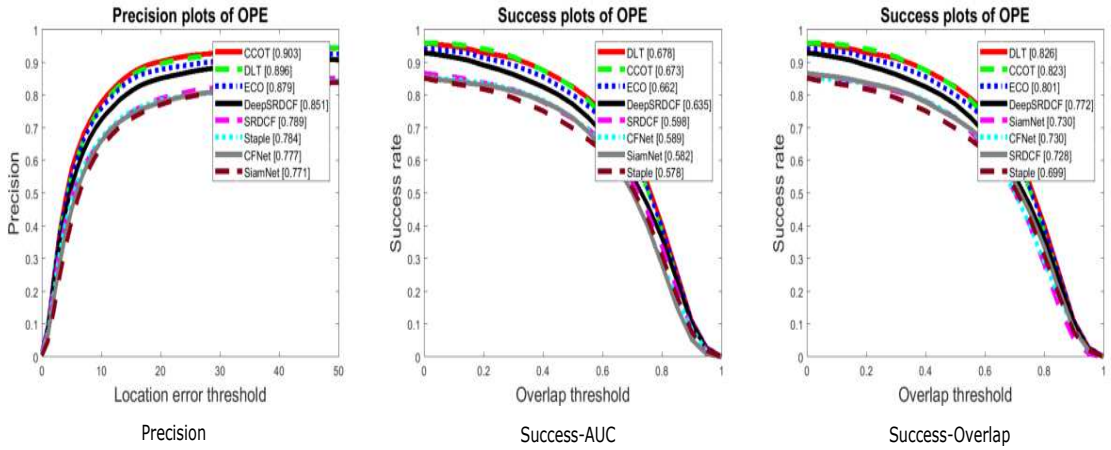


Fig. 5.4 Qualitative results of our tracker on selected frames of 4 benchmark sequences, where the robustness of DLT (red box) in handling tracking failures of ECO (green box) tracker can be clearly observed. The sequences are selected based on various challenges that occur within them.

5.3.3 Evaluation on the object tracking benchmark (OTB100)

The OTB100 benchmark dataset provides 100 ground-truth-annotated sequences. The frames have annotations of specific challenges such as illumination variation, scale variation, occlusion, motion blur, deformation, in-plane and out-of-plane rotations, out of view, fast motion, background clutter, and low resolution. One Pass Evaluation (OPE) on OTB100 compares the proposed DLT tracker to the best state-of-the-art trackers in terms of overlap precision, the area under the curve (AUC) score, and success plots, as shown in Figure 5.5. By 1.93% in overlap precision, 2.41% in AUC score, and 3.12% in success rates, the DLT tracker ($precision = 0.896, AUC = 0.678, success = 0.826$) outperforms the base



Precision Success-AUC Success-Overlap

Fig. 5.5 Comparison of DLT with state-of-the-art trackers on OTB100 in terms of distance precision, AUC score and overlap success using one-pass evaluation (OPE). The proposed DLT has significant improvements over the base ECO tracker in all the cases.

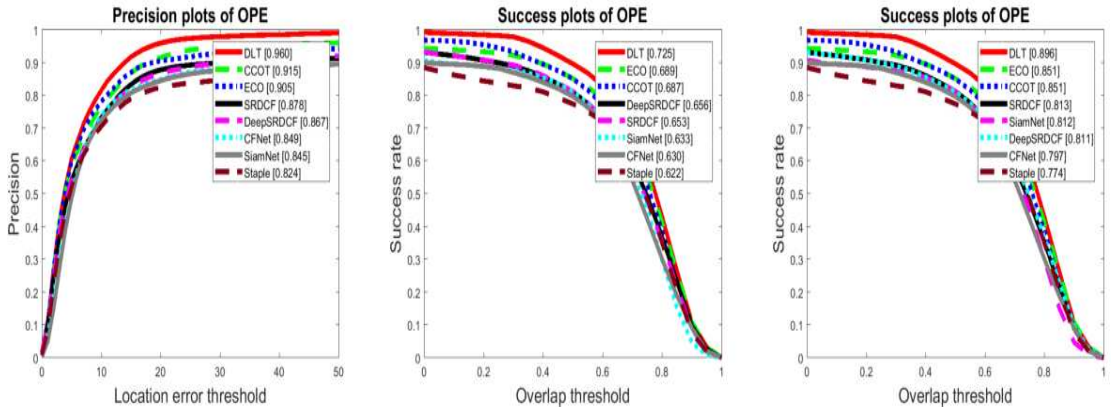
ECO tracker ($precision = 0.879, AUC = 0.662, success = 0.801$). In addition, our tracker outperforms all other state-of-the-art trackers in terms of success rates and ranks second in terms of precision and AUC score.

5.3.4 Evaluation on the object tracking benchmark 50 (OTB50)

The OTB50 benchmark dataset comprises 50 ground-truth-annotated sequences. Similar to OTB100, frame annotations for individual challenges are included. In One Pass Evaluation (OPE) on OTB50, the proposed DLT tracker is compared to the top state-of-the-art trackers in terms of overlap precision, AUC score, and success plots, as shown in Figure 5.6. The DLT tracker ($precision = 0.960, AUC = 0.725, success = 0.896$) outperforms the base ECO tracker ($precision = 0.905, AUC = 0.689, success = 0.851$) in overlap precision, AUC, and success rates by 6.07%, 5.22%, and 5.28%, respectively. The proposed DLT tracker outperforms the state-of-the-art trackers in all the three evaluation metrics.

5.3.5 Evaluation on the unmanned aerial vehicle (UAV123) dataset

The proposed DLT tracker is tested using a 123-video aerial video benchmark dataset. Precision, AUC score, and overlap success are used to perform a state-of-the-art comparison. As demonstrated in Table 5.1, the DLT tracker surpasses the compared state-of-the-art trackers, gaining 1.6% in precision, 1.5% in AUC score, and 1.73% in success rates over the baseline tracker.



(a)Precision (b)Success-AUC (c)Success-Overlap

Fig. 5.6 Comparison of DLT with state-of-the-art trackers on *OTB*50 in terms of distance precision, AUC score and overlap success using one-pass evaluation (OPE). The proposed DLT has significant improvements over the base ECO tracker in all the cases.

Table 5.1 Results for tracker evaluation on the UAV123 dataset. The precision, AUC and success rates are reported. The proposed DLT tracker outperforms the compared state-of-the-art trackers in terms of all the evaluation metrics.

	DLT	**ECO**	**SRDCF**	**MUSTER**	**STRUCK**	**CSK**	**KCF**	**TLD**
Precision	0.762	0.750	0.676	0.591	0.578	0.488	0.466	0.439
AUC	0.536	0.528	0.464	0.391	0.381	0.311	0.297	0.283
Success	0.647	0.636	0.551	0.455	0.413	0.329	0.317	0.294

5.3.6 Evaluation on the visual object tracking dataset

The VOT dataset [82] contains 60 video sequences, each with one or more of the five labels of motion change, illumination change, occlusion, size change, and camera motion annotated on each frame. Table 5.2 compares the proposed detection-based long-term (DLT) tracker to the top-ranked trackers on the VOT dataset regarding Expected Average Overlap (EAO) and Robustness. The top three trackers' values are displayed in red, blue, and green. Over the base ECO tracker, the failure rate has decreased by 15.46%, and the average overlap has increased by 4.1%, demonstrating the efficacy of the proposed DLT algorithm in improving the robustness of correlation filter based trackers.

5.3.7 Attribute based evaluation

The trackers are analysed based on their ability to track in a range of challenging situations. The attribute level analysis demonstrates the effectiveness of our detector-based approach in re-initializing the base tracker after a target loss. Thus, the proposed technique allows

Table 5.2 Results for tracker evaluation on the VOT dataset. The average overlap and failure rate are reported. Our DLT tracker obtains the best performance among the compared trackers on this dataset.

Tracker	DLT	ECO	CCOT	Staple	DSR DCF	SR DCF	Color KCF	KCF	DSST	Struck
EAO	0.3709	0.3563	0.3310	0.2952	0.2763	0.2471	0.2262	0.1924	0.1814	0.1416
Failure rate	0.82	0.97	0.89	1.42	1.23	1.43	1.50	1.95	2.38	3.40

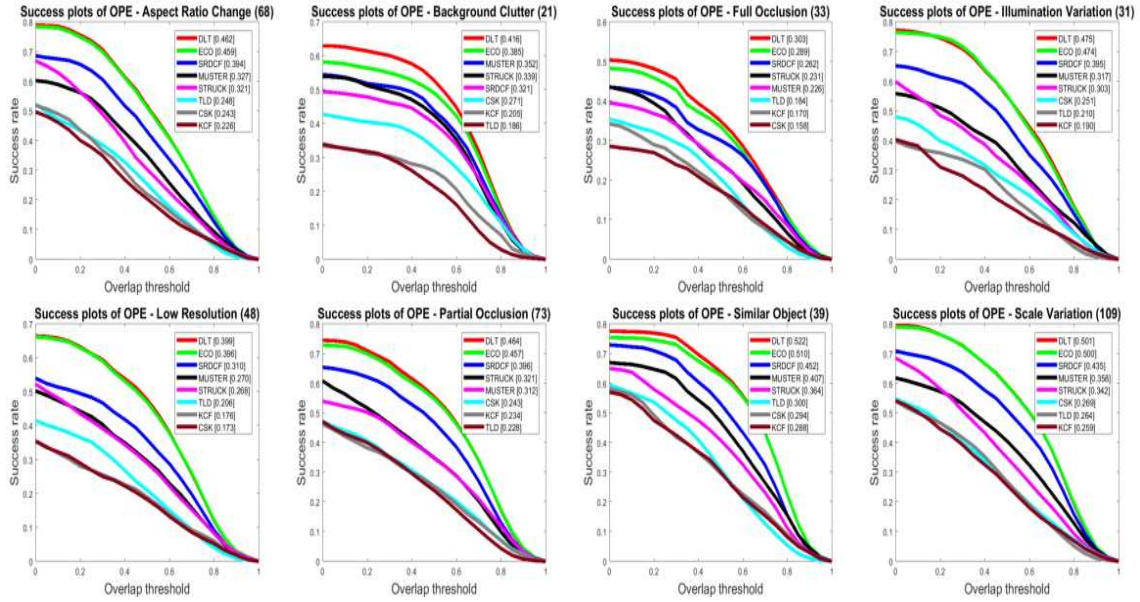


Fig. 5.7 Success plots on the UAV123 dataset for eight different attributes- aspect ratio, background clutter, full occlusion, illumination variation, low resolution, partial occlusion, similar object and scale variation. The area-under-the-curve scores for the state-of-the-art trackers are shown in the legend.

the tracker to track for long duration under various challenging conditions such as fast motion, background clutter, deformation, illumination variation, in-plane and out-of-plane rotations, scale variations, occlusions, out-of-view, aspect ratio change, similar object, and low resolution. Figures 5.7 and 5.8 demonstrate attribute level analysis for various attributes on UAV123 and OTB50, respectively.

5.4 Chapter Summary

This chapter detailed a long-term tracking algorithm by including a detector in the fundamental correlation filter-based tracker framework. The essential contribution is to create a detector using a parts-based BoVW technique and to use online learning to incorporate

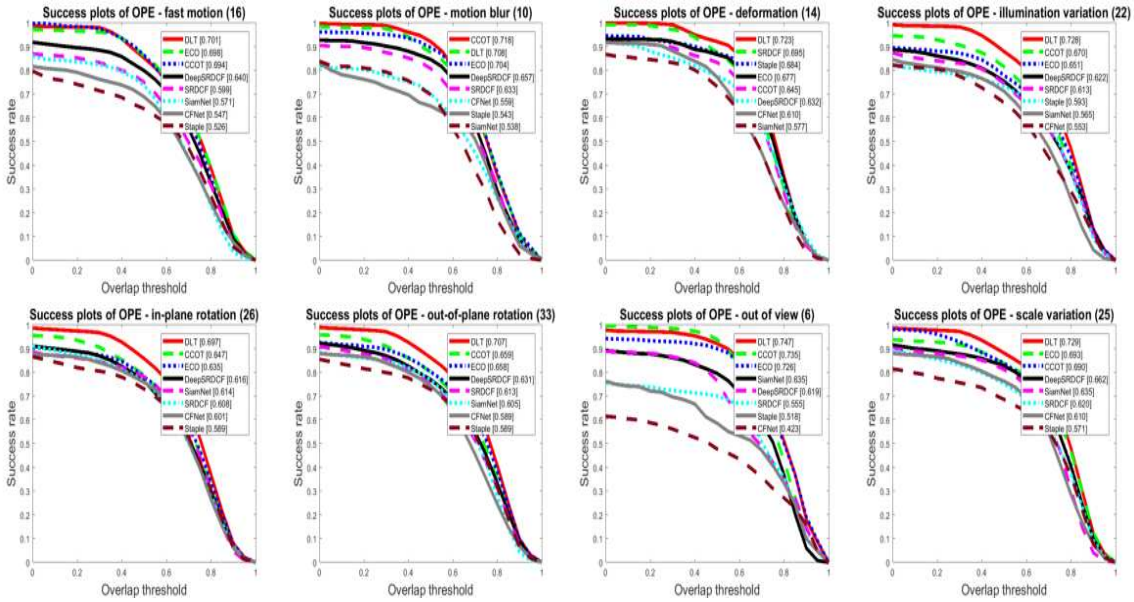


Fig. 5.8 Success plots on the OTB dataset for eight different attributes- fast motion, motion blur, deformation, illumination variation, in-plane rotation, out-of-plane rotation, out of view and scale variation. The area-under-the-curve scores for the state-of-the-art trackers are shown in the legend.

temporal information into the detector. The proposed detector-tracker integration is particularly efficient because the detector initiates only when the track loses. The existing integration allows the tracker to restart, otherwise impossible in most current state-of-the-art correlation filter trackers. On thorough analysis of the performance of the DLT algorithm it was identified that model drift in CFTs were caused by another major factor - the recursive search technique around the previous location. This limitation is addressed by a new tracking paradigm discussed in Chapter 6.

Chapter 6

SACOT: Segmentation guided Attention for Correlation based Object Tracking

The model drift in correlation filter trackers can be minimized by tracking resumption as discussed in Chapter 5. The general approach of object tracking algorithms is to execute a multi-scale search around the previous target location. Suppose the learned model fails to track the target in any frame, the recurrent search around the wrong location causes the model to deviate significantly from the actual path, following the background, as seen in Figure 6.1. The traditional search technique is therefore unreliable and is a major reason for model drift. It is critical to find regions of interest with high confidence for target localization.

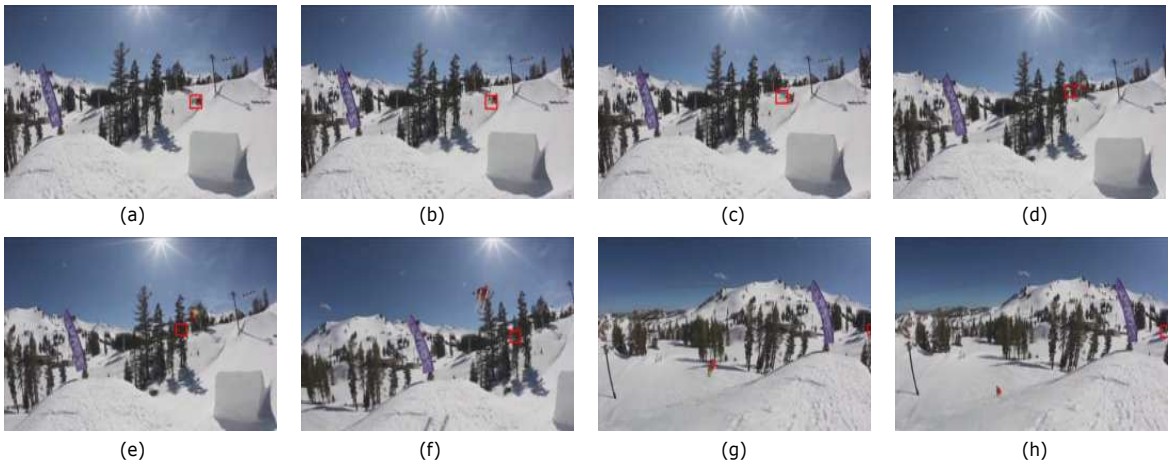


Fig. 6.1 Model drift in a conventional tracker [2] caused by the recursive search around previous target location after a tracking failure. (a) The tracker locates the target object. (b) The tracker loses the target object and learns the wrong target appearance. (c) - (h) The tracker lost the target and deviates entirely from the actual target path. The red box denotes the target located in each frame.

A generic technique of segmentation-guided attention, integrable into any correlation filter based tracking framework, is proposed to generate highly confident search regions for tracking. Segmentation masks can serve as a guiding step to locate the target in the tracking scenario, as evidenced by the benefits of segmented object masks and the gradual substitution of rectangular bounding boxes with segmented masks in the recent tracking benchmarks [88]. Compared to the commonly used rectangular bounding boxes [3, 62–64], which, when loosely specified, may capture the background and the target, causing errors in the tracking process, segmentation masks have better localization capabilities by providing pixel-wise labelling of the objects.

A novel single-object tracking framework is developed, which is made up of two modules and is led by segmentation-based attention: the Region Proposal Module (RPM) provides a coarse location of the target through object proposals and the Target Localization Module (TLM) predicts the final target state, represented by a bounding box x, y, w, h, where (x, y) is the target center and (w, h) is the target width and height, respectively. A segmented object mask generated by the widely used instance segmentation method Mask-RCNN (MRCNN) [89] extracts the search region for target estimation. A correlation filter learned from the first frame is used to perform target localization on the generated regions of interest. A domain adaptation technique adds target-level information into the tracking pipeline, establishing target-specific learning corresponding to each video sequence.

The contributions of the proposed method can be summarised as follows:

1. This study addresses a segmentation-guided attention method for single object tracking, proposed utilizing a correlation filter tracker-driven region proposal module (RPM) and target localization module (TLM).

2. The proposed segmentation guided tracking is developed in two generic variants: SACOT and Domain Adaptive SACOT (DA-SACOT) (without and with domain adaptation, respectively)

3. The region proposal module uses an initial offline training. The localization module learns the target appearance through an online model update to properly include domain-specific knowledge into the tracking pipeline.

4. Extensive evaluations are conducted and the proposed methods achieved state-of-the-art performance on benchmark tracking datasets with a significant average gain of 2.47% in precision, 2.55% in AUC score and 2.15% in overlap score through domain adaptation across the evaluated datasets.

6.1 Related works

This section gives a comprehensive overview of the current methodologies used in fields that are closely related to our work. The proposed tracking framework identifies and addresses the constraints of each type of method.

6.1.1 Correlation based trackers

CF tracker research focuses on feature representation, optimization, and the ability to deal with specific challenges. The primary feature representations employed in current trackers are handcrafted features [61] and deep convolutional features [37, 90]. Regression is used to integrate the global context into the CF trackers in [83] and [42]. [91] proposed an efficient multi-cue analysis framework for reliable visual tracking. In contrast, [92] proposed a novel adaptive spatially regularised correlation filters model that simultaneously optimizes the filter coefficients and the spatial regularisation weight. Despite advances in research, object variations and environmental factors continue to limit the performance of CF trackers. As a result, the proposed method uses segmented object masks to incorporate attention guidance into the correlation tracker, which, when combined with the online model update methodology [61], significantly increases tracking accuracy.

6.1.2 Detection based trackers

In the tracking scenario, tracking-by-detection techniques have primarily been employed to deal with missed detections. In [85], a random fern classifier-based online monitoring system is proposed. In [93], researchers offer a long-term tracking framework based on a Siamese Region Proposal Network (RPN) based re-detector and an offline trained meta-updater that incorporates temporal variations of the object into tracking. In [94], a visual object tracking by detection approach is proposed, in which the target object is detected by examining the forward and backward trajectories from numerous correlation filter trackers. [95] discusses a Convnet design that conducts detecting and tracking simultaneously. [96] discusses a vehicle tracking algorithm that uses a detection-based tracking framework. Missed detections limit detection-based trackers when tracking a previously undetected object type. The bounding box detections may not be precise enough to define the object, resulting in tracking failures due to background clutter. The proposed method employs segmentation masks that better characterize a target object to guide tracking and an initial offline learning approach to bring target-level knowledge into the tracking model.

6.1.3 Segmentation based trackers

There are two types of segmentation-based tracking approaches: bottom-up methods and hybrid methods, according to [97]. Bottom-up approaches begin by segmenting the foreground parts of the frames, extracting features from these segmented regions, and then tracking these features. Contour matching [98–100] picks the best candidate image patch that matches a shape-based object appearance model, and contour propagation [101, 102] uses a state-space model to move the object contour to a new location in the considered frame. The main disadvantage of this method is that the final tracking accuracy is dependent on the accuracy of features retrieved from the segmented foreground. Therefore, segmentation errors can also lead to tracking inaccuracies. [103–105] are hybrid algorithms that fully leverage the relationship between object tracking and foreground segmentation. These methods combine segmentation and tracking into a probabilistic or graph-based framework and use iterative energy-minimization algorithms, to extract object masks.

6.1.4 MRCNN based trackers

The MRCNN has grown in popularity as a benchmark for image instance segmentation, and it has recently been extended to video instance segmentation. The joint segmentation and tracking approaches are employed in the multi-object tracking domain, where MRCNN segments all instances in a frame, and a data association technique is used to link objects between frames. MaskTrack RCNN [106] adds a new branch to MRCNN and uses external memory to monitor the target across frames to accomplish object detection, tracking, and segmentation. MaskProp [107] extends MRCNN by adding a mask propagation branch that propagates instance masks between frames. Track-RCNN [108] uses 3D convolutions and an association head to link the objects across the frames to bring time information into MRCNN.

6.2 Proposed Methodology

A single-object tracking framework driven by segmentation is proposed to focus on highly confident search regions rather than the traditional technique of a blind recursive search around the previous location to reduce model drift in tracking algorithms. The suggested solution consists of an offline fine-tuned Region Proposal Module (RPM) and an online learned Target Localization Module (TLM). The MRCNN architecture serves as the foundation for the RPM. The TLM augments a correlation filter learned independently from the first frame to localize the target instances in each frame. RPM uses Regions of Interest (ROIs) to narrow down the exhaustive search space for the target object and then applies the

learned correlation filter to the RoIs to determine the final target bounding box. RPM and correlation filters share the backbone features derived from a pre-trained Resnet model [109]. Using target specific information from each video sequence, a simple yet effective fine-tuning of the RPM is done. During tracking, the video frames are processed consecutively in an online manner. Figure 6.2 depicts the of the general framework of the proposed method.

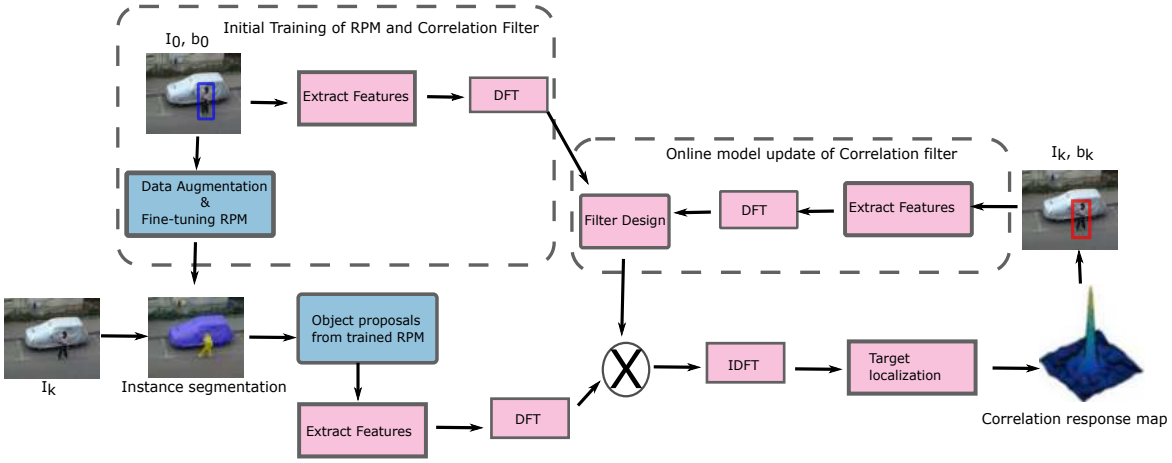


Fig. 6.2 The overall framework of the proposed segmentation guided visual object tracking. The initial domain adaptive training of both RPM and CF are done from first frame I_0 where the initial bounding box b_0 is known. From $k = 2,..N$, each frame I_k is input to the fine-tuned RPM which performs instance segmentation and generates object proposals. The learnt Correlation Filter (CF) acts on the object proposals and locates the final target location b_k. CF adopts an online model update using features from k^{th} and $(k-1)^{th}$ frames. The correlation filter performs all operations in the Fourier domain.

6.2.1 The Region Proposal Module (RPM)

In general, tracking algorithms assume that the target object's frame-to-frame mobility is limited, and hence search frequently about the previous target center to detect the target in each frame. When the tracker fails in an intermediate frame, repeated search in successive frames can produce a drift in the predicted target route. The solution to the model drift caused by the incorrect search space is to establish a more confident search space.

It is proposed in this work to build a search region around the previous location using object segmentation. The MRCNN framework is used for segmentation, and it is trained offline to add target-specific knowledge into the tracking pipeline. In Section 6.2.3, the domain adaptive training of MRCNN is described in detail. Figure 6.3 illustrates how the RPM works on an example sequence from OTB100 [70].

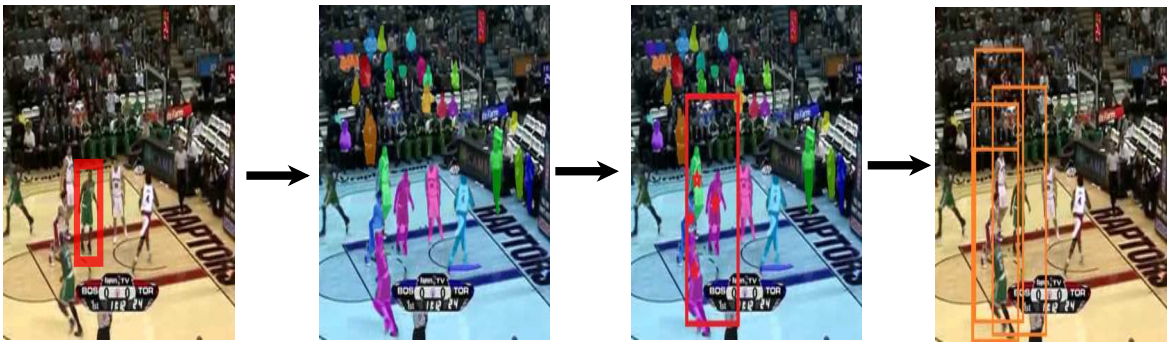


Fig. 6.3 Working of RPM: (a) Input frame with the groundtruth target shown in a red box (b) Instance segmentation using target adaptive MRCNN on the input frame (c) A search area three times the target size is selected (shown in red box) and segmented instances within the search area are the candidate locations (centers shown in red $*$) (d) The object proposals extracted around the candidate locations (rectangular areas of three times the target size, shown in orange boxes).

Instance segmentation is performed using the target-specific MRCNN, which generates pixel-wise masks for multiple object instances in each frame. The segmented instances within a three-times-target-size rectangular area around the previous location are chosen. As a result, the nearest instances are the most likely candidates for target estimation. A search region is then extracted around each candidate location, a rectangular area three times the initial target size. The search space size is determined by trial and error to strike a compromise between missing segmentations and processing costs. The candidate object proposals are represented by these rectangular patches, which are then processed by the target localization module.

The RPM has two distinct advantages:

1. The segmented region proposals have a higher chance of being chosen as the target.

2. Target estimate considers a larger search field, which aids in locating objects with larger displacements produced by quick motion.

If no segmented instances are formed in a frame due to the limitations of the RPM network, the search region surrounding the previous target location is deemed the object proposition.

6.2.2 Target Localization Module (TLM)

The RPM can create possible target positions but cannot distinguish between the target object and background noise. As a result, the RPM includes a localization module whose

primary goal is to carry out this distinction. Unlike RPM, the target localization module is trained entirely online to build a response map. The target localization module uses a correlation filter to compute a rough $2D$ target position in each frame and provide robustness by minimizing pseudo detections.

Training the model

To learn a classifier f with d channels, the TLM employs a set of training data $\{(x_k, y_k)\}_{k=1}^{n}$ with n samples created from the first frame. The training samples x_k are produced by circular shifting of the original target patch [61], with y_k being the intended Gaussian response map of the same size as the extracted feature map. The correlation of a sample x_k with the first target is defined by the response map. The correlation filter learns by reducing the L^2- error between the actual response map y_k and the obtained response map on the training samples x_k [61],

$$\min_{f^l} \sum_{l=1}^{d} ||x^l * f^l - y||^2 + \lambda ||f^l||^2 \tag{6.1}$$

where f^l and x^l signify the respective channels of the learned correlation filter and feature map, $*$ denotes circular convolution, and $\lambda \geq 0$ denotes the regularization term.

Online Tracking

The target is localized in subsequent frames using the correlation filter learned in the first frame. The learned correlation filter, f, is applied channel-wise on the feature map derived from each test sample, creating a series of response maps $\{(S_i(z))\}_{i=1}^{n_o}$ in the spatial domain as:

$$S_i(z) = F^{-1}\left(\sum_{l=1}^{d} \hat{z}_i^l \bigodot \hat{f}_i^l\right) \tag{6.2}$$

The element-wise multiplication is denoted by $\bigodot$, the Discrete Fourier Transform (DFT) is denoted by $\wedge$, and the inverse DFT is denoted by F^{-1}.

The final target localization

The general rule is to choose the location of the maximum score when given a correlation response map. Many similar objects or a background similar to the target may exist in the object tracking scenario, leading to multiple areas in the correlation map with equal maximum scores. To avoid false detections, it is suggested that instead of maximising $s(u, v)$

alone, maximise $\frac{s(u,v)}{||(u-u*,v-v*)||}$ across n_o response maps. The newly calculated target centre is (u,v), while the previous frame's target centre is (u^*,v^*). Eventually, the object with the highest correlation score and the smallest deviation from the previous target position is found.

The functioning of target localization on an example sequence from OTB50 is depicted in Figure 6.4.

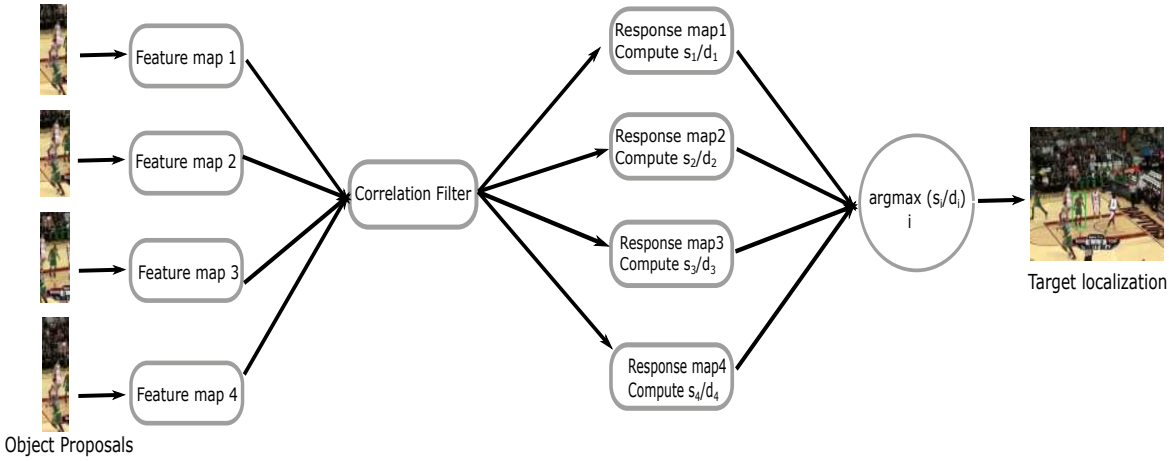


Fig. 6.4 Working of the target localization module: Feature maps are extracted from the object proposals received from RPM. The learned correlation filter generates response maps corresponding to each extracted feature map. The score to distance ratio, $\frac{s_i}{d_i}, i = 1,...n_o$, across n_o response maps (here, $n_o = 4$) is maximized to estimate the final target location.

6.2.3 Domain Adaptation Strategy

A simple yet successful technique incorporates target-specific knowledge from each video sequence into the proposed tracking framework: an offline fine-tuning of the RPM and an online model update of the TLM.

Fine tuning the RPM

The computing demands and a lack of training data limit the online learning of RPM from each frame. Data augmentation is used in the initial frame to provide enough training data from the target sample. The generated training data is used in the initial offline training of the RPM network.

The following geometric transformations, image filtering, and image enhancement processes are used to generate 100 versions of the initial target sample:

1. Bounded rotations with an interval of 20° in the range $(20,360)$.

2. Unbounded rotations with an interval of 20° in the range $(20, 360)$.
3. Scaling in the range $(0.5, 2)$ at 0.25 intervals across the width and height.
4. Mean, Gaussian and Median blurring using kernels of size $(3, 5, 7)$.
5. OpenCV libraries were used to improve sharpness, contrast, and brightness, as well as histogram equalisation.

With Resnet [109] trained on Imagenet (1000 object classes) [110] as the backbone network, the RPM network is initialised with MRCNN weights learned from the COCO segmentation dataset [111]. Due to the limited number of training samples, offline learning can only fine-tune the final network heads for a single epoch. With low computing complexity, this learning method introduces target-specific knowledge into the RPM network.

Generating groundtruth masks for training

The benchmark tracking datasets and evaluation toolkits [62–64] use a rectangular representation of the target object, represented by x, y, w, h, where (x, y) is the target center and (w, h) is the target width and height. For object representation, several toolkits [3] have recently started to use segmentation masks. Because rectangular groundtruths are used in most datasets, and the MRCNN model requires segmented groundtruth for training, it is necessary to automate the creation of groundtruth masks from the available rectangular representations.

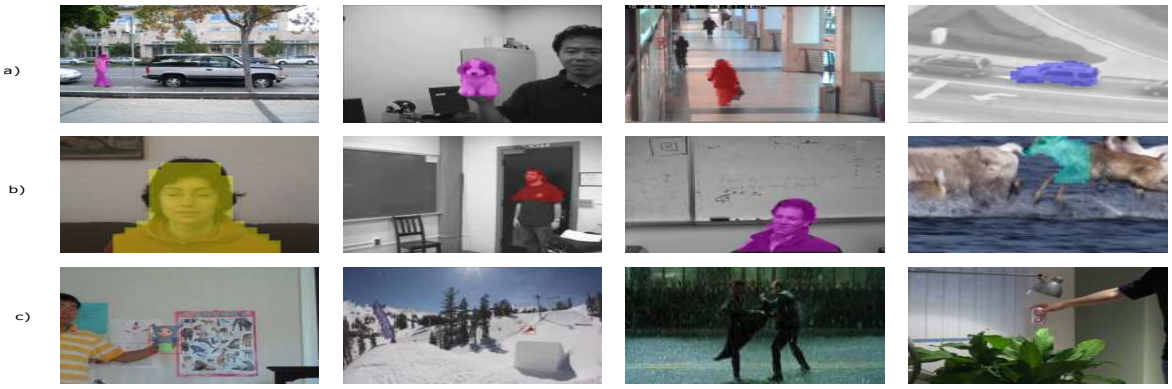


Fig. 6.5 Groundtruth masks generated from rectangular bounding boxes using MRCNN on first frames of selected benchmark sequences. a) Perfect masks b) Perfect object masks with some background segmentation. c) No masks generated. For the no mask case the filled boxes can be used as masks.

The MRCNN model is used to build groundtruth masks from rectangular boxes because of its accuracy and to reduce the computational overhead of a new segmentation approach. In

the first frame, a rectangular region with a width of $3 \times w$ and a height of $3 \times h$ is extracted around the target centre and fed into the MRCNN to generate segmented masks. To ensure that no pixel from the groundtruth is lost, an auxiliary region is fed. Three possible scenarios are analysed based on qualitative evaluation on benchmark datasets, as indicated in Figure 6.5.

Fig. 6.6 The qualitative results of applying the domain adapted MRCNN, fine-tuned with target specific features, on selected frames of benchmark sequences is shown. Each row depicts frames from a sequence with the target object in the sequence segmented. The original pre-trained MRCNN model (without the proposed domain specific fine-tuning) fails to detect the target objects in these sequences, clearly indicating the advantage of the proposed domain adaptation strategy.

1. Perfect masks: The segmented groundtruth mask is ideally generated by MRCNN from the rectangular bounding box.

2. Masks with background information: The target object is accurately segmented by MRCNN, but some extraneous background is segmented as well. When the target of

interest is merely the head/face, as illustrated in the second row of Figure 6.5, MRCNN segments some extra background.

3. No segmentation: MRCNN is unable to detect any object.

The first two instances can be taken care by picking only pixels within the original bounding box, whereas the third instance requires filling the tight rectangular bounding box. As shown in Figure 6.5, the error in the third instance will be negligible, with very few extra background pixels. Figure 6.6 shows the qualitative results of domain adaptive RPM on selected frames of benchmark sequences.

Correlation filter update

The correlation filter is updated online using target samples from each frame to handle the appearance variations of the target in a video sequence. On every frame, the feature map x and the correlation filter f are changed as follows:

$$\hat{\mathrm{x}}^t = (1-\alpha)\hat{\mathrm{x}}^{t-1} + \alpha\hat{\mathrm{x}}^t \tag{3a}$$

$$\hat{f}^t = (1-\alpha)\hat{f}^{t-1} + \alpha\hat{f}^t \tag{3b}$$

where α is the learning rate and t is the index of the current frame.

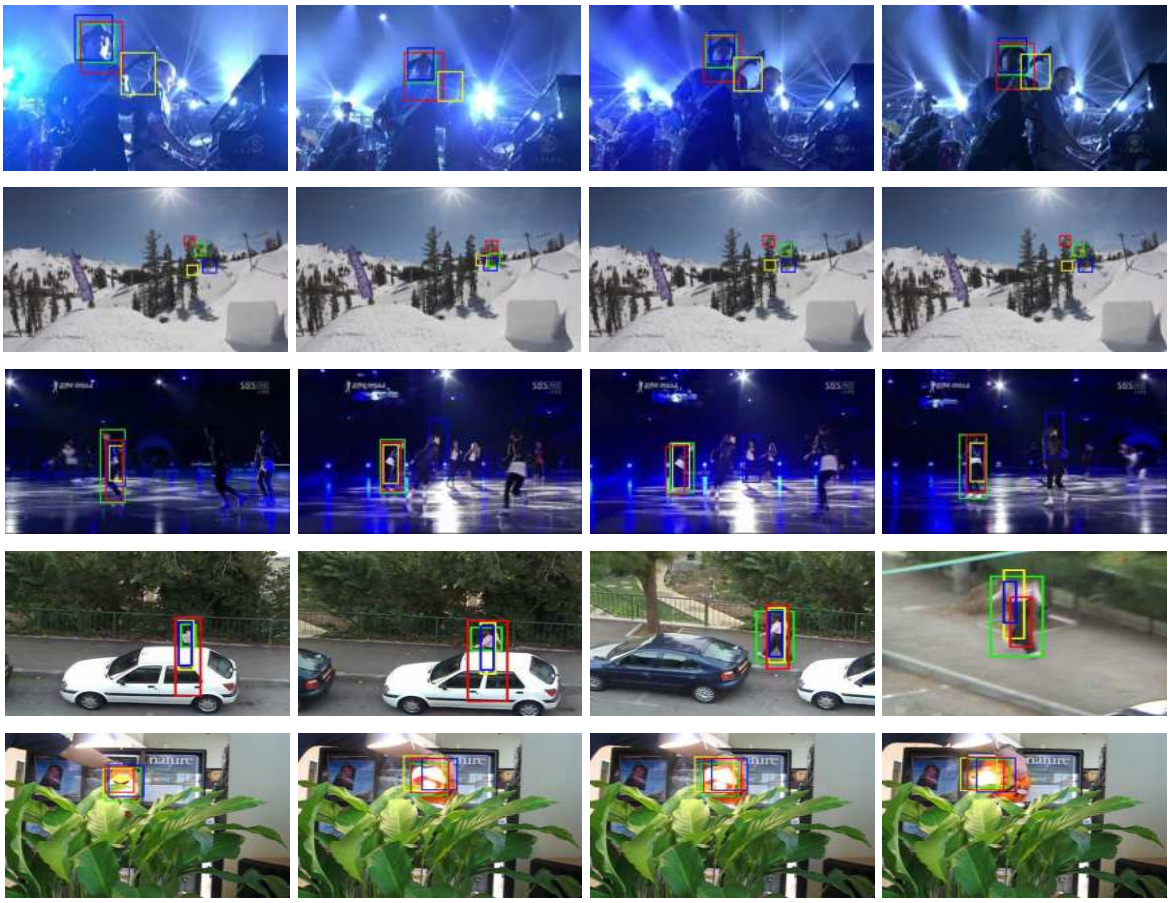
6.2.4 Comparison with Segmentation based trackers

Considering the advantages and disadvantages of both bottom-up and hybrid segmentation-based tracking approaches outlined in Section 6.1.3, a novel solution is proposed to combine the two challenges of segmentation and tracking. Rather than relying entirely on the segmentation mask, the proposed method uses segmentation as a guidance tool to localize the target object, addressing the limitations of block-based methods. As with hybrid approaches, a correlation filter tracker is learned to use the correlation between segmentation and tracking completely. This tracker is then used to correctly estimate the target location in the segmented region. The correlation filter tracker's simplicity and speed overcome the computational penalty in target localization that hybrid approaches encounter.

6.2.5 Comparison with MRCNN based trackers

The proposed method uses a correlation filter to maintain track of the target path in each frame, extending the MRCNN model into a single object tracking framework. The MRCNN-

based approaches presented in Section 6.1.4 have yet to be completely investigated in the single object tracking benchmark datasets, hence they are not available for state-of-the-art comparison.



UDT [112] **ROAM [113]** **SRDCF [2]** **DA-SACOT (Proposed)** **(as indicated by color coding of bounding boxes)**

Fig. 6.7 Qualitative comparison of the proposed DA-SACOT tracker (green) with SOTA methods. The selected frames are attributed by several tracking challenges like similar objects, low resolution, viewpoint changes, occlusion, background clutter, scale changes, etc. The proposed method better localizes the target object compared to the other selected methods.

6.3 Experimental Results

6.3.1 Implementation Details

The proposed tracker is implemented in Python using Tensorflow, with all evaluations on a 12*GB* NVIDIA Tesla K80 GPU. The RPM backbone is a pre-trained MRCNN model

learned on the COCO dataset, and the correlation filter extracts target features using Resnet pre-trained on Imagenet. DA-SACOT uses the same parameter settings as in [89] to fine-tune the RPM. The learning rate α for the online model update in the TLM is set to 0.02, following the pattern of general CF trackers [61]. The proposed tracker's average online tracking speed is 11 frames per second (fps).

6.3.2 Qualitative Results

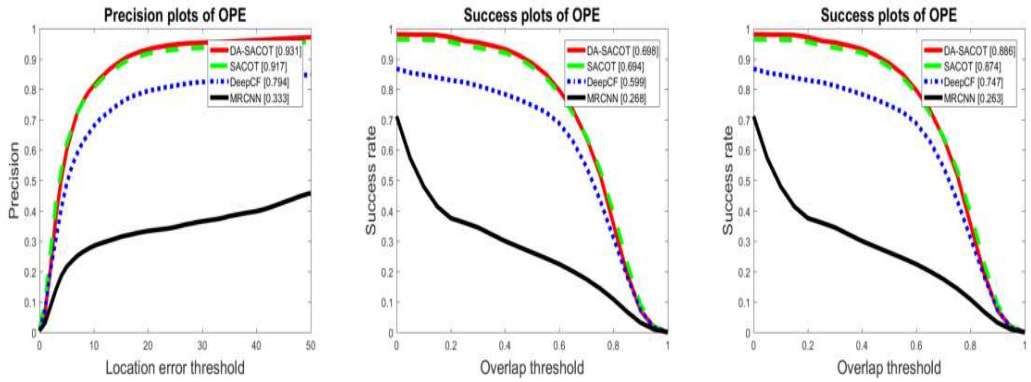
The qualitative findings of the proposed DA-SACOT tracker on benchmark sequences are discussed in this section. The bounding boxes estimated are compared to those generated using state-of-the-art methods. Figures 6.7 and 6.8 illustrate a qualitative comparison of the proposed tracker on selected frames having various challenges such as scale variations, low resolution, background clutter, occlusion, viewpoint changes, rotation, etc. Compared to other competing methods, the bounding boxes created by the proposed method are more localized, proving the algorithm's success in target adaptation and model drift reduction.

SiamBAN [114] TADT [115] ASRCF [92] GCT [116] DA-SACOT (Proposed) (as indicated by color coding of bounding boxes)

Fig. 6.8 Qualitative comparison of the proposed DA-SACOT tracker (green) with SOTA methods. The selected frames are attributed by several tracking challenges like rotations, fast motion, viewpoint changes, occlusion, deformation, background clutter, etc. The proposed method better localizes the target object compared to the other selected methods.

6.3.3 Ablation Study

The ablation investigations on the OTB50 dataset are described here in detail to determine the role of each component of the DA-SACOT tracker in increasing overall tracking performance. The proposed model is upgraded in stages from the baseline components to the full version, demonstrating the effectiveness of each key element. Individual tracking performance of the fundamental building blocks: i) MRCNN and ii) deep correlation filter (DeepCF), as well as two variants: SACOT and DA-SACOT, of the proposed method are contrasted in this section.



(a) Precision Score (b) AUC Score (c) Overlap Score

Fig. 6.9 Ablation studies for each component of the proposed tracking algorithm on OTB50. (a) Overall precision score, (b) Success score using AUC and (c) Success score using overlap ratio. MRCNN denotes the application of pre-trained MRCNN on OTB sequences, DeepCF denotes a correlation filter tracker learned from deep features extracted from pre-trained Resnet features, SACOT and DA-SACOT are the proposed methods without and with domain adaptation respectively. The scores obtained in each method is shown in the legend.

The evaluation procedure is detailed below:

1. **MRCNN**: On each frame the pre-trained MRCNN model of the RPM is applied, and the final target location is selected as the detection closest to the previous target location.

2. **DeepCF**: The target localization module is formed by the correlation filter tracker learned from Resnet features and is evaluated on every frame.

3. **SACOT**: The proposed Segmentation guided correlation filter tracking algorithm is assessed, which integrates the region proposal and target localization modules.

4. **DA-SACOT**: The domain-adapted SACOT method is evaluated here, fine-tuned from each video sequence utilising target-specific features.

The overall precision, Area Under the Curve (AUC), and overlap scores obtained from the one-pass evaluation (OPE) are shown in Figure 6.9, and Table 6.1 shows the attribute-based evaluation of four different variants on the OTB50 dataset under conditions of deformation (DEF), occlusion (OCC), out-of-plane-rotation (OPR), and low resolution (LR). Over the baseline correlation filter (DeepCF), the suggested segmentation guided tracking (SACOT) improves precision, AUC, and overlap by 15.49%, 15.85%, and 17.00%, respectively. This demonstrates the benefit of using highly confident segmented search regions to locate a target. Using the suggested target-specific learning, the precision, AUC, and overlap scores are improved by 1.53%, 0.58%, and 1.37%, respectively.

Table 6.1 Attribute level comparison of the individual components and the proposed methods on OTB50. The compared attributes are deformation (DEF), occlusion (OCC), out-of-plane rotations (OPR) and out-of-view (OV). The precision and success scores clearly indicate the contribution of each component to the overall tracking performance.

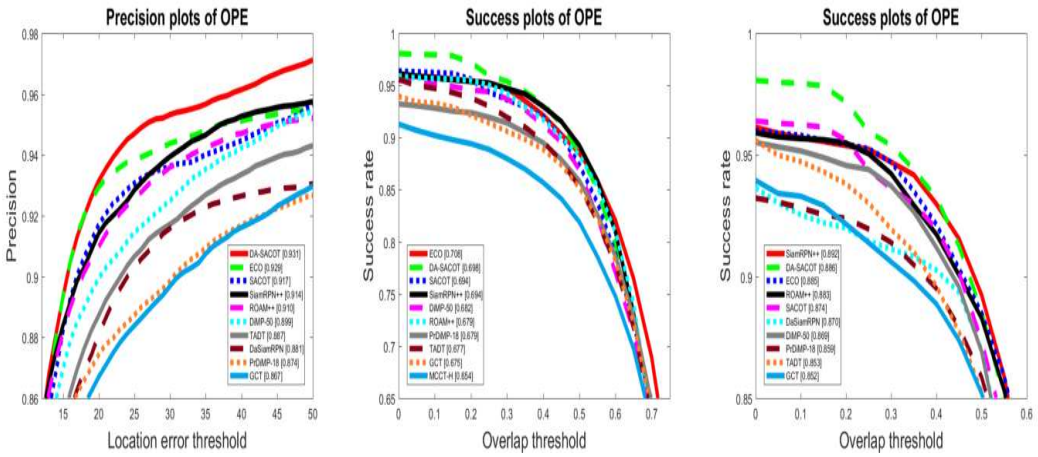
	Precision				**Success**			
	DEF	**OCC**	**OPR**	**LR**	**DEF**	**OCC**	**OPR**	**LR**
MRCNN	0.383	0.422	0.282	0.223	0.301	0.290	0.240	0.186
DeepCF	0.819	0.730	0.799	0.407	0.613	0.566	0.586	0.342
SACOT	0.924	0.928	0.920	0.753	0.683	0.703	0.689	0.576
DA-SACOT	0.968	0.955	0.941	0.749	0.713	0.717	0.698	0.577

6.3.4 Evaluation on the Object Tracking Benchmark (OTB50)

The proposed approach is evaluated on 50 videos from OTB50. Over a range of overlap thresholds, the precision and success rates of the proposed SACOT and DA-SACOT trackers are compared to eight existing competing trackers [10, 117, 113, 118, 115, 119, 120, 116]. As shown in Figure 6.10, the proposed segmentation guided tracker (SACOT) performs comparably to them, and the proposed domain adaptation approach (DA-SACOT) increases SACOT's reported accuracy by 1.53%, 0.58%, and 1.37% in precision, AUC, and overlap scores, respectively.

6.3.5 Evaluation on the Unmanned Aerial Vehicle (UAV123) Dataset

Figure 6.11 compares nine recent competing approaches to the quantitative results on the aerial video benchmark dataset UAV123 (which consists of 123 videos with an average of 915 frames) on the same metrics as OTB. The precision, AUC, and overlap scores of the SACOT algorithm are 82.1%, 76.0%, and 62.2%, respectively, which are incredibly similar



(a) Precision Score (b) AUC Score (c) Overlap Score

Fig. 6.10 Comparison of the proposed methods - Domain Adaptive - SACOT (DA-SACOT) and SACOT with recent trackers on OTB50 in terms of (a) distance precision, (b) AUC score and (c) overlap success using one-pass evaluation (OPE). For clarity in presentation, the axes' limits are trimmed, removing the overlapping plots. The average scores used to rank the trackers is shown in the legend.

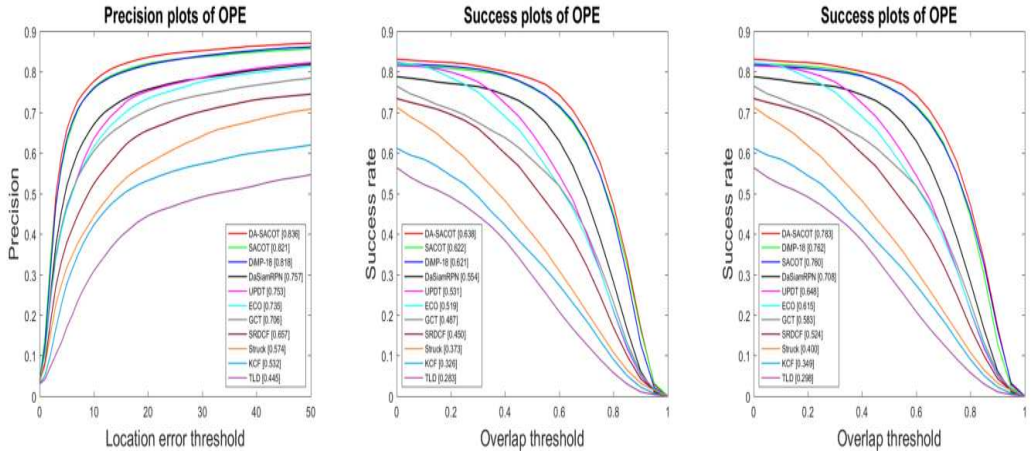
to the latest high-performing tracking methods [2], [61], [10], [118],[119], [116], [121], [122], [60]. In terms of accuracy, AUC, and overlap scores, domain adaptive fine-tuning increases the SACOT by 1.83%, 3.03%, and 2.57%, respectively.

6.3.6 Evaluation on the TempleColor (TC128) Dataset

With 128 color videos, the Temple-Color is a more difficult benchmark. As demonstrated in Figure 6.12, the proposed SACOT tracker achieves equivalent performance compared to 10 state-of-the-art trackers [2],[37],[91],[10],[115],[34],[112], with a precision score of 81.4%, an AUC score of 75.1%, and an overlap score of 59.6%. Compared to the suggested SACOT without domain adaptation, the domain adaptive - SACOT (DA-SACOT) improves precision by 4.05%, AUC score by 4.26%, and overlap by 2.52%.

6.3.7 Attribute based evaluation

Fast motion (FM), background clutter (BC), deformation (DEF), out-of-plane rotations (OPR), scale variations (SV), partial occlusions (PO), total occlusions (OCC), out-of-view (OV), aspect ratio change (ARC), viewpoint changes (VC), and low resolution are some of the tracking challenges that the proposed trackers are put to (LR). This category analysis



(a) Precision Score (b) AUC Score (c) Overlap Score

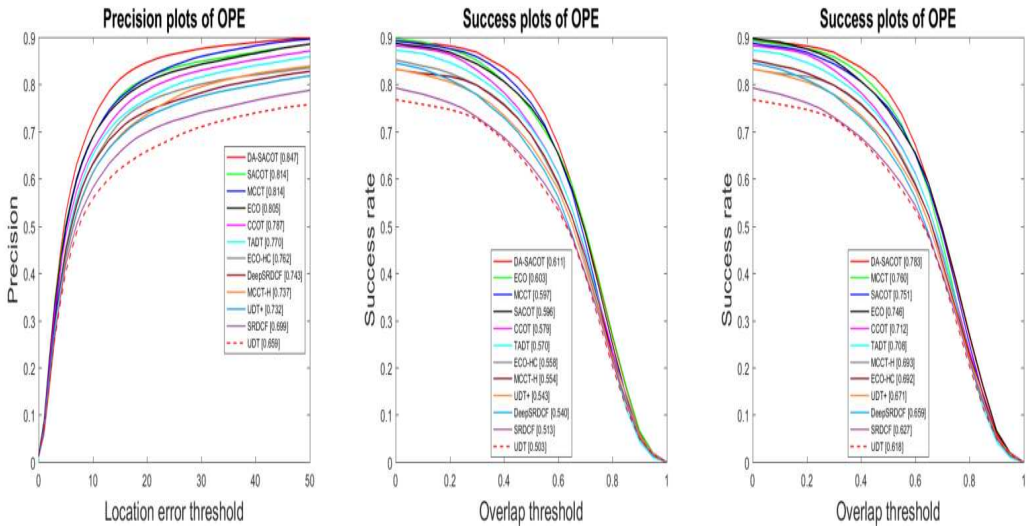
Fig. 6.11 Comparison of the proposed SACOT and DA-SACOT with recent competing trackers on UAV123 in terms of (a) distance precision, (b) AUC score and (c) overlap success using one-pass evaluation (OPE). The proposed method outperforms the other trackers in all the cases.The scores obtained in each method is shown in the legend.

demonstrates the effectiveness of the proposed segmentation guided tracking in achieving accurate and long-term tracking. The attribute level comparison on the OTB50, UAV123, and TC128 datasets are summarised in Tables 6.2, 6.3 and 6.4, with the top three ranks shown in red, green, and blue colors, respectively. Under all the challenges considered, the proposed SACOT and DA-SACOT trackers give top performance.

Table 6.2 Attribute level comparison of proposed SACOT and DA-SACOT trackers on OTB50. The compared attributes are deformation (DEF), occlusion (OCC), out-of-plane rotations (OPR) and out-of-view (OV). The top three trackers under each category are shown column wise in red, green and blue respectively.

	Precision				**AUC**				**Overlap**			
Trackers	**DEF**	**OCC**	**OPR**	**OV**	**DEF**	**OCC**	**OPR**	**OV**	**DEF**	**OCC**	**OPR**	**OV**
DA-SACOT (Ours)	0.968	0.955	0.941	0.912	0.713	0.717	0.698	0.725	0.894	0.916	0.873	0.885
SACOT (Ours)	0.924	0.928	0.920	0.919	0.683	0.703	0.689	0.736	0.855	0.894	0.857	0.899
ECO [10]	0.910	0.960	0.935	0.953	0.683	0.722	0.701	0.756	0.884	0.912	0.868	0.957
SiamRPN++ [117]	0.905	0.873	0.934	0.891	0.675	0.656	0.697	0.705	0.896	0.846	0.897	0.887
RetinaNet_CF	0.927	0.890	0.876	0.885	0.665	0.664	0.653	0.680	0.847	0.841	0.860	0.879
DiMP-50 [118]	0.921	0.862	0.909	0.815	0.683	0.657	0.673	0.658	0.868	0.831	0. 861	0.811
DaSiamRPN [119]	0.915	0.907	0.885	0.815	0.669	0.669	0.653	0.677	0.903	0.815	0.868	0.797
GCT [116]	0.909	0.886	0.879	0.815	0.685	0.639	0.673	0.647	0.886	0.810	0.854	0.829
PrDiMP-18 [120]	0.884	0.830	0.898	0.795	0.661	0.643	0.681	0.651	0.853	0.875	0.868	0.784
MCCT-H [91]	0.873	0.841	0.893	0.779	0.672	0.643	0.659	0.637	0.851	0.853	0.823	0.760

The superior performance of the proposed method can be attributed to the following factors: i) A domain-specific learning mechanism that learns target-specific knowledge both



(a) Precision Score (b) AUC Score (c) Overlap Score

Fig. 6.12 Comparison of the proposed SACOT and DA-SACOT trackers with recent competing trackers on TC128 in terms of (a) distance precision, (b) AUC score and (c) overlap success using one-pass evaluation (OPE). The SACOT shows comparable SOTA performance and the proposed method using domain adaptation (DA-SACOT) outperforms the other trackers in all the cases. Scores obtained in each method is shown in the legend.

Table 6.3 Attribute level comparison of the proposed SACOT and DA-SACOT trackers on UAV123. The compared attributes are aspect ratio changes (ARC), out of view (OV), partial occlusion (PO) and viewpoint changes (VC). The top three trackers under each category are shown column wise in red, green and blue respectively.

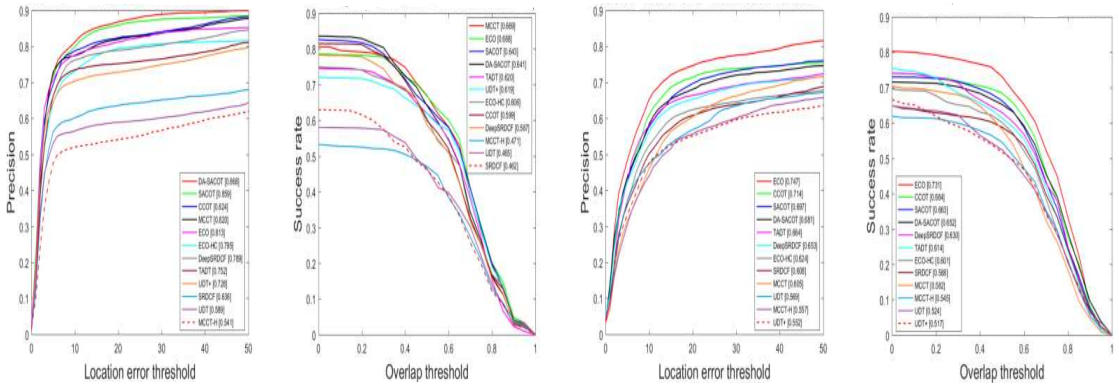
	Precision				**AUC**				**Overlap**			
Trackers	**ARC**	**OV**	**PO**	**VC**	**ARC**	**OV**	**PO**	**VC**	**ARC**	**OV**	**PO**	**VC**
DA-SACOT (Ours)	0.835	0.799	0.754	0.892	0.626	0.595	0.546	0.701	0.770	0.704	0.670	0.855
SACOT (Ours)	0.794	0.744	0.734	0.826	0.590	0.559	0.522	0.644	0.726	0.666	0.641	0.780
DiMP-18 [118]	0.802	0.754	0.727	0.811	0.607	0.576	0.533	0.641	0.752	0.699	0.656	0.787
PrDiMP-18 [120]	0.800	0.724	0.731	0.822	0.612	0.567	0.540	0.654	0.753	0.692	0.667	0.801
DaSiamRPN [119]	0.737	0.582	0.649	0.743	0.528	0.455	0.458	0.567	0.673	0.559	0.582	0.717
UPDT [121]	0.707	0.559	0.671	0.712	0.472	0.399	0.450	0.502	0.540	0.434	0.542	0.575
ECO [10]	0.652	0.514	0.656	0.670	0.441	0.396	0.444	0.474	0.480	0.440	0.523	0.519
GCT [116]	0.648	0.552	0.626	0.672	0.420	0.370	0.415	0.468	0.486	0.407	0.490	0.546
SRDCF [2]	0.572	0.498	0.578	0.583	0.382	0.359	0.368	0.411	0.402	0.387	0.421	0.433
Struck [122]	0.491	0.426	0.477	0.512	0.313	0.294	0.304	0.339	0.288	0.245	0.334	0.331
KCF [61]	0.445	0.432	0.477	0.458	0.265	0.252	0.283	0.290	0.252	0.212	0.313	0.289
TLD [60]	0.407	0.378	0.375	0.419	0.252	0.252	0.222	0.277	0.257	0.243	0.243	0.283

online and offline, and ii) A segmentation approach that generates a highly confident search space. The attribute level study confirms that the proposed trackers can effectively deal with model drift induced by various tracking problems. However, as illustrated in Figure 6.13,

Table 6.4 Attribute level comparison of the proposed SACOT and DA-SACOT trackers on TC128. The compared attributes are background clutter (BC), fast motion (FM), low resolution (LR) and scale variations (SV). The top three trackers under each category are shown column wise in red, green and blue respectively.

	Precision				AUC				Overlap			
Trackers	**BC**	**FM**	**LR**	**SV**	**BC**	**FM**	**LR**	**SV**	**BC**	**FM**	**LR**	**SV**
DA-SACOT (Ours)	0.824	0.804	0.840	0.819	0.591	0.592	0.610	0.613	0.754	0.772	0.773	0.775
SACOT (Ours)	0.764	0.788	0.815	0.781	0.565	0.589	0.597	0.597	0.702	0.755	0.744	0.740
MCCT [91]	0.789	0.754	0.803	0.825	0.585	0.569	0.588	0.622	0.747	0.730	0.742	0.791
ECO [10]	0.726	0.783	0.777	0.760	0.562	0.598	0.588	0.590	0.683	0.751	0.708	0.717
CCOT [34]	0.716	0.764	0.793	0.766	0.541	0.576	0.590	0.573	0.648	0.729	0.716	0.685
TADT [115]	0.686	0.718	0.792	0.724	0.516	0.546	0.574	0.568	0.607	0.669	0.705	0.687
MCCT-H [91]	0.684	0.704	0.722	0.716	0.530	0.544	0.539	0.566	0.646	0.665	0.658	0.700
ECO-HC [10]	0.674	0.730	0.721	0.701	0.515	0.548	0.530	0.539	0.627	0.688	0.630	0.648
DeepSRDCF [36]	0.637	0.695	0.709	0.670	0.473	0.517	0.519	0.521	0.556	0.635	0.624	0.626
UDT+ [121]	0.614	0.698	0.671	0.620	0.480	0.527	0.500	0.484	0.567	0.645	0.589	0.582
SRDCF [2]	0.605	0.659	0.701	0.694	0.458	0.496	0.524	0.525	0.546	0.609	0.623	0.641
UDT [121]	0.560	0.595	0.633	0.643	0.457	0.468	0.492	0.502	0.536	0.582	0.584	0.610

the proposed technique has additional scope for improvement in circumstances of camera motion and illumination variations.



(a) Precision(CM) (b) Success(CM) (c) Precision(IV) (d) Success(IV)

Fig. 6.13 Limitations of the proposed domain adaptive tracker under conditions of (a)-(b) camera motion (CM) and (c)-(d)illumination variations (IV) is shown in terms of precision and success scores on the OTB50 dataset.

6.3.8 The impact of Domain adaptation technique

On all three datasets, the Table 6.5 summarises the overall gain in accuracy, AUC, and overlap scores obtained through domain adaptation. This demonstrates the value of embedding target-specific knowledge into the proposed tracking system via domain adaptation.

Table 6.5 Relative gain in DA-SACOT compared with SACOT obtained through domain adaptation

	OTB13			UAV123			TC128		
	Precision	**AUC**	**Overlap**	**Precision**	**AUC**	**Overlap**	**Precision**	**AUC**	**Overlap**
SACOT	0.917	0.694	0.874	0.821	0.760	0.622	0.814	0.751	0.596
DA-SACOT	0.931	0.698	0.886	0.836	0.783	0.638	0.847	0.783	0.611
Gain(%)	1.53	0.58	1.37	1.83	3.03	2.57	4.05	4.26	2.52

6.4 Chapter Summary

This chapter presents a general segmentation-driven single object tracking framework to minimize model drift in conventional trackers. The application of domain adaptation and segmentation for visual object tracking is one of the essential components of the proposed approach. Image segmentation builds highly confident search regions by leveraging the localization benefits of segmented object masks. In addition, a domain adaptation technique, both online and offline, induces target-specific knowledge into the tracking pipeline. The effectiveness of the segmentation-guided attention mechanism in tracking is demonstrated by accurate target estimation and a lower failure rate, according to both qualitative and quantitative analyses. The proposed tracker demonstrated significant increases across various challenging scenarios, including background clutter, fast motion, object rotations, low resolution, deformations, occlusion, aspect ratio changes, viewpoint changes, and out-of-view. However, the bottleneck of the proposed method is the need for performance improvements in conditions of camera motion and illumination variations. Considering the limitations of the DA-SACOT, the thesis finally explores the use of visual attention in correlation filter tracking framework. The consequent proposal is detailed in Chapter 7.

Chapter 7

Co-Attention Maps for Discriminative Object Tracking (CAM-DOT)

7.1 Introduction

A popular and effective tracking paradigm of the current time is correlation filter tracking, in which the target is estimated by the cross-correlation between the template image and a search region. A multi-channel rectangular template represents the target. Modern template-based trackers locate targets using an effective brute-force search. This approach works well for transformations like translation, scale changes, etc. Still, it is ineffective in more general circumstances, such as those that result in aspect ratio and rotation changes. Modern trackers integrate sampling, bounding box refinement/regression networks, and approximate exhaustive search as a compromise for aspect ratio estimation. These methods are limited to rectangles aligned along one axis.

Bounding boxes are inaccurate approximations of the target, which makes them unreliable for estimating high-dimensional template-based transformation [54]. This is typical; for instance, consider long, rotating, deformable objects or a human with outstretched hands. Under these circumstances, a binary per-pixel segmentation mask is the most precise and well-defined target representation. In short-term tracking scenes [123], when the target only occupies a portion of the image, significantly changes its appearance over a longer duration, and moves across a crowded background, top video object segmentation techniques perform poorly. The most effective trackers use visual model adaptation. However, when segmentation errors occur, this results in an unrecoverable tracking failure [124].

In general, the targets may experience significant changes in their visual appearance and the available settings may be cluttered and distracting, causing long-duration tracking

difficult. However, the human visual system never had trouble tracking. Research in psychology and cognitive science suggests that human perception is selective and attention based. An object will draw our attention once it has entered the camera's frame of view so that we can concentrate on how it moves. Motion has a significant impact on the tracking process. Human attention is drawn to movement, and tracking results from attention. It is widely acknowledged that the human visual system depends on visual attention. Applications including target location, image retrieval, camera auto-focus, and image compression have all used attention models on static images.

While tracking robustness has increased dramatically over the past few years, tracking precision has not kept pace. The issue of precise target estimate has received less attention because of the focus on the creation of strong classifiers. Most trackers use a straightforward multi-scale search to estimate the target bounding box. We contend that this method is fundamentally constrained because target estimate is a challenging task requiring in-depth object comprehension. The proposed approach integrates correlation and visual attention on adjacent frames to enable accurate localization of the target object throughout a video sequence, generating object representations in the form of bounding boxes and segmentations. The proposed approach relies on an end-to-end trained co-attention network to generate co-attention maps that highlight the locations of shared salient objects in a set of images. The co-attention network computes the similarity between the input images using an encoder-decoder design combined with a mutual correlation layer. To locate the final target, the segmentation prediction network in D3S [125] receives the co-attention map, correlation response map, and feature map of the current frame. The performance of the proposed CAM-DOT tracker surpasses that of the recent state-of-the-art trackers, highlighting the benefit of co-attention maps in object tracking.

7.2 Proposed Methodology

This work proposes a novel tracking approach consisting of three components: 1) An offline learned target estimation module based on co-attention, 2) An online learned target classification module based on a correlation filter, and 3) a pre-trained discriminative segmentation network.

The Resnet-18 model, trained on Imagenet, is the backbone network for both target estimation and classification. A co-attention network performs target estimation, which is trained offline on an extensive object cosegmentation dataset [126] adapted from the PASCAL dataset. The weights of the co-attention network are frozen during online tracking. The co-attention network correlates the current frame with a reference pool of previously processed

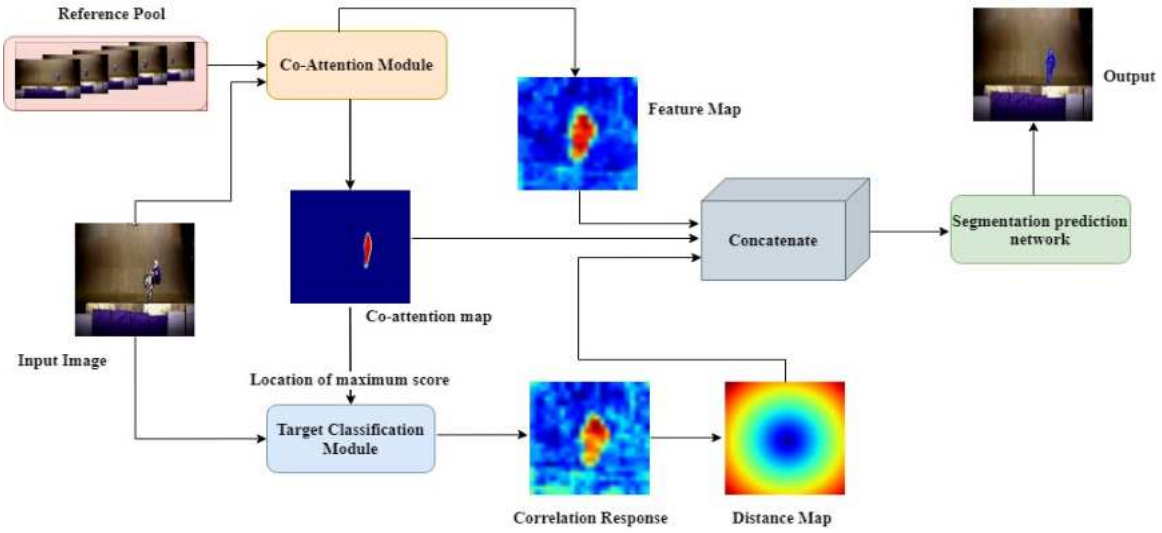


Fig. 7.1 The overall framework of the proposed CAM-DOT tracker. The co-attention module correlates the current frame with a reference pool to generate the co-attention map. The location of highest correlation in the response map is used as the search location for target classifier. The distance map generated from the classifier, the co-attention map highlighting the common object, and the feature map of the current frame are integrated in the segmentation network to generate the final segmented target.

frames. It generates a segmented co-attention map that highlights the predominantly common salient object between the search image and reference images. A deep formulation of the discriminative correlation filter as in ATOM [127] performs target classification. The discriminative classifier is learned online using target features extracted using the backbone network. The classifier predicts a target correlation score in each frame to differentiate the target object from other objects in the scene. To generate the final segmented target, the CAM-DOT tracker employs the pre-trained segmentation network in D3S [125], fed with the co-attention map, correlation response map, and backbone features, all generated from the current frame.

The proposed framework separates the subproblems of target estimation and classification. Both tasks are later integrated into a pre-trained discriminative segmentation network to localize the target location in each frame, as shown in Figure 7.1.

7.2.1 The Co-attention Network for Target Estimation

The CAM-DOT tracker relies on an end-to-end trained co-attention network to generate the co-attention map highlighting the regions of common objects between any two images. A Siamese network can be effectively used to match two images. Therefore, the proposed

co-attention network is built on a straightforward yet effective Siamese encoder-decoder architecture. Figure 7.2 illustrates the architecture of the proposed co-attention network, which consists of three components: (a) A pair of Siamese encoder networks extract feature maps, f_1 and f_2, from the input images, (b) the Correlation Layer computes the semantic correlation between the extracted feature maps to generate the correlation heatmap, CM, and (c) a Decoder network generates the final co-attention map through deconvolution operations on the extracted features and correlation heatmap. In the following sections, the three components of the co-attention network are detailed, along with the training procedure.

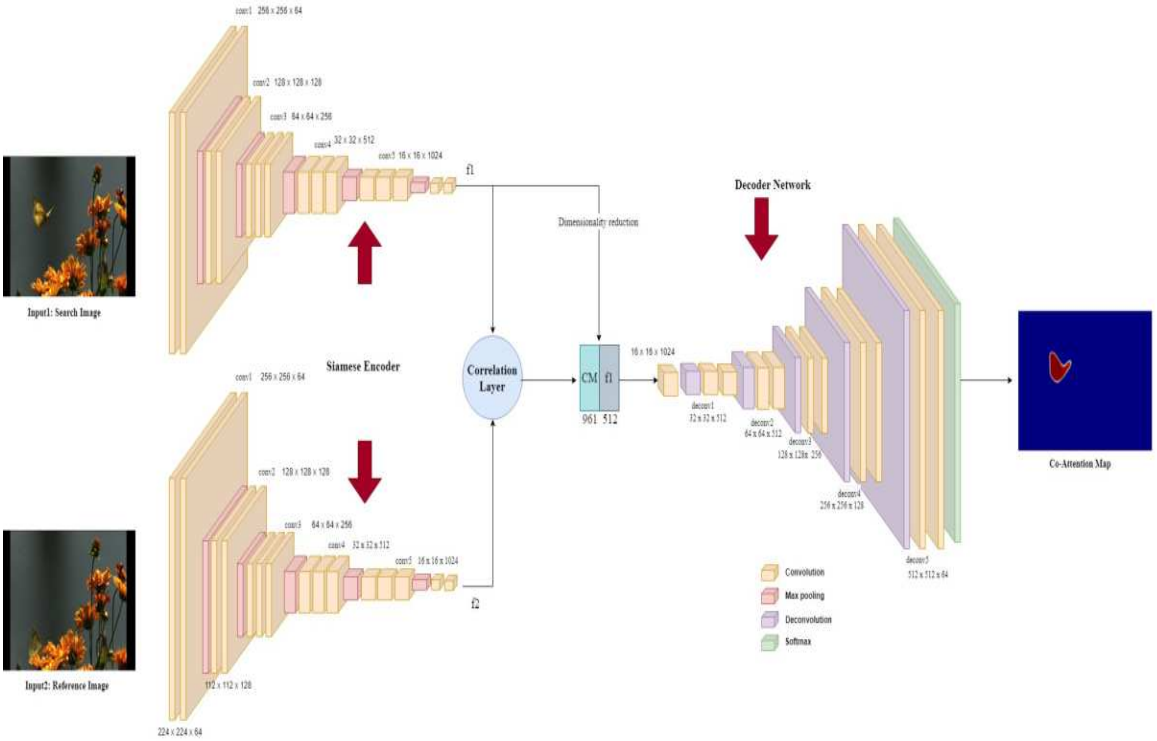


Fig. 7.2 The co-attention network includes three parts: (i) a Siamese encoder network that generates feature maps of the input images, (ii) a correlation network that generates correspondence map *CM* by feature matching, and (iii) the concatenation of feature maps and the correspondence map are passed through a Siamese decoder network to obtain the common object mask.

The Siamese Encoder

A Siamese encoder with two identical feature extraction convolutional neural networks that share parameters makes up the first part of the co-attention network. Each encoder network is based on the VGG16 architecture [128], with two 3×3 convolutional layers in place of

the final two fully connected layers to produce feature maps with greater spatial detail. As a result, each encoder has five pooling layers and fifteen convolutional layers. The Siamese encoder pair accepts input images of 512×512 pixels, producing two feature maps, f_1 and f_2, each with 1024 channels and a 16×16 spatial resolution.

The Correlation Layer

The feature maps, f_1, and f_2, obtained from the Siamese encoder represent the high-level semantic content of the input images. Two images with the same object will contain similar features at the locations of the common object. Finding the locations of common objects can be aided by computing the correlation between each pixel in one image and each other pixel in the other image. The feature points of the two input images are thus matched by the co-attention network using a correlation layer. Through a pixel-by-pixel comparison of the two feature maps, f_1 and f_2, the correlation layer filters the high-level features of the target image to emphasize the locations of common objects. The pixel-wise correlation between the feature vectors $f_1(i,j)$ and $f_2(m,n)$ is computed as:

$$CM(i,j,k) =< f_1(i,j), f_2(m,n) > \tag{7.1}$$

Where $k = (n-j)D + (m-i)$ and $D \times D$ is patch size. The common object can be at any location within the image; hence, the patch size is set to $D = 2 * max(w-1, h-1) + 1$, where w and h are the width and height of the feature maps f_1 and f_2, respectively. The output correlation map, CM, is of size $w \times h \times D^2$.

Decoder

The third component of the co-attention network is the decoder network, which generates the final co-attention map. The feature map f_2 of the target image is concatenated with the correlation map, CM, and input to the decoder. Five blocks make up the decoder, each having two convolutional layers before a deconvolutional layer. The decoder employs the ReLU activation function following each convolution and deconvolution. The final co-attention map, which is the same size as the input image, is generated by a Softmax function applied at the final layer.

Training the Co-attention network

The task of creating co-attention maps is formulated as a binary image labelling problem, which is then utilized for training the co-attention network using the standard cross entropy

loss function. The cosegmentation dataset used in [126], which was in turn modified from the PASCAL dataset [129], is utilized for training the CAM-DOT tracker. Twenty classes of foreground objects and one type of background object make up the dataset. It has $2,857$ pixel-level labelled images for validation and $8,498$ training images. As a training set for co-segmentation, $161,229$ pairs of images with similar objects were selected from the training images. The object class labels are ignored in favour of labelling the common objects as foreground since the intention is to draw attention to the common object from the pair of images.

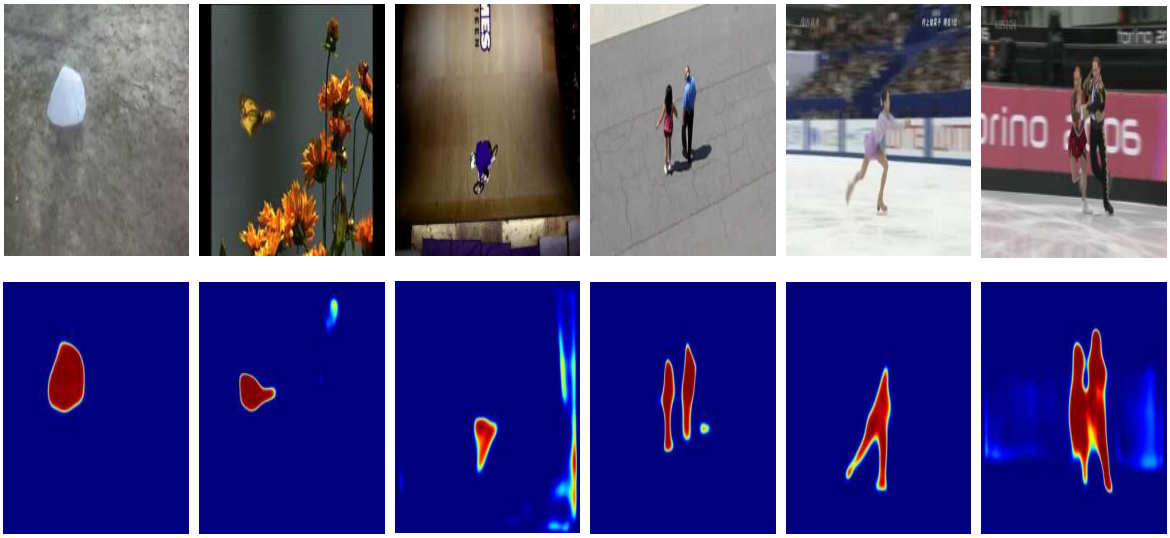


Fig. 7.3 Sample co-attention maps generated on sample frames of the VOT dataset [3].

Adaptation to target appearance variations

In object tracking, the appearance of the target varies throughout the video sequence. Hence, there is a high probability of missing the target if the co-attention network matches the target image with only the first frame or the immediately previous frame. The CAM-DOT tracker uses a reference pool of images to handle such appearance changes by generating a group co-attention map. The reference pool is randomly constituted of five previously tracked frames. Let the reference pool be $R = R_1, ..., R_n$. Each reference image R_i is paired with the target image and input to the trained co-attention network to predict the co-attention map for all the five pairs of images, $C = c_k, 1 <= k <= K$, where c_k is the predicted co-attention map of the k^{th} image pair. The final co-attention map, CM, highlighting the target object in the current frame is computed as:

$$CM(x,y) = mean\{c_k(x,y)\} \tag{7.2}$$

The sample co-attention maps generated by the trained co-attention network is shown in Figure 7.3.

7.2.2 The Deep correlation filter for Target Classification

The co-attention network distinguishes the target from the background and estimates its location precisely, but it cannot discriminate the target from similar objects. Since discriminative correlation filters have the well-established ability to locate a target robustly, the CAM-DOT tracker employs a recent deep formulation of the correlation filters for target classification. The position of the highest response in the correlation map produced by the deep DCF is the most likely target location, much like in the general correlation filter trackers. The final component of the CAM-DOT tracker, the segmentation network, requires a probability of the target at each pixel location. Therefore, the generated correlation map is transformed into a distance map using the Euclidean distance from the location of maximum response to the remaining pixels.

7.2.3 Final Target Localization

The co-attention and distance maps generated from the first two components provide complementary information about the target probability at each pixel location. While the distance map gives a more reliable position estimate of the object, the co-attention map provides more details on the target but is less discriminative. To generate the final segmented target estimate, CAM-DOT relies on the pre-trained segmentation network in D3S [125]. This network combines the co-attention map, distance map, and the features from the backbone network and upscales them to generate a segmentation map of the same size as the input image. Finally, the segmentation network creates a rotational bounding box from the segmented object mask using a bounding box fitting technique.

7.2.4 Object tracking with CAM-DOT

As stated in Section 7.2.1, the end-to-end pre-trained network serves as the co-attention network in CAM-DOT. The first five frames are added to the reference pool of the Siamese encoder. After the fifth frame, the reference pool is composed of randomly chosen previously tracked frames. By employing a region four times the target size and the initialization process outlined in [127], the deep correlation filter is learned from the first frame. For each additional frame considered during online tracking, a search region four times the target size is extracted. The Siamese encoder uses this search region as its target image, and it correlates

it with the reference pool to create the co-attention map. The deep DCF builds the distance map using this search space, as explained in Section 7.2.2. The segmentation network then calculates the bounding box and final target segmentation, as described in Section 7.2.3. The reference pool is randomly updated every ten frames to tackle the target appearance changes. The deep DCF is then updated using the new target estimate following [127]. Figure 7.4 shows the sample target masks and bounding box estimates of the CAM-DOT tracker on sample sequences of VOT dataset [3].

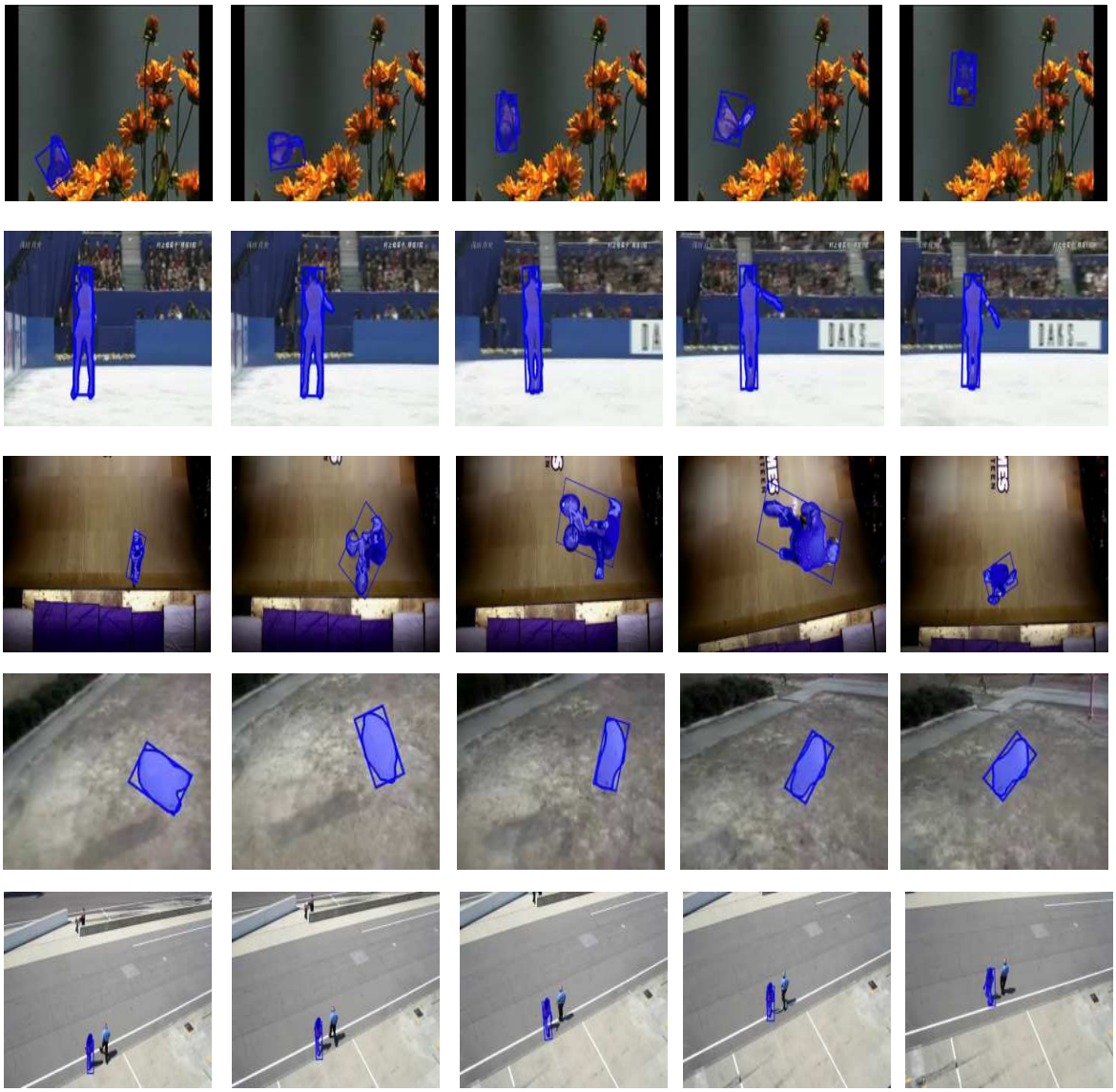


Fig. 7.4 The segmented target estimate and bounding boxes generated by the CAM-DOT tracker in sample frames of VOT sequences.

7.3 Experimental Results

7.3.1 Evaluation on OTB100 dataset

The proposed CAM-DOT tracker is evaluated on 100 videos from OTB. Over a range of overlap thresholds, the precision and success rates of the proposed tracker is compared to eight existing competing trackers [130, 131, 118, 132, 93, 127, 125]. As shown in Figure 7.5, the proposed CAM-DOT outperforms the recent top trackers in terms of precision, success (AUC) and overlap ratio.

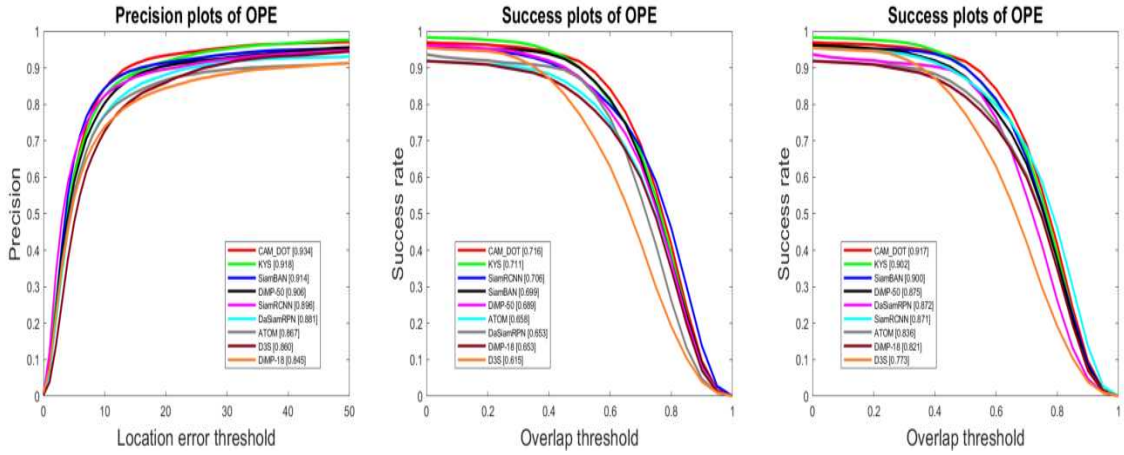


Fig. 7.5 OTB100 - state-of-the-art comparison

7.3.2 Evaluation on VOT dataset

The VOT dataset comprises of 60 sequences. Targets are labelled with rotated rectangles to allow for a better examination compared to other datasets. According to the VOT evaluation process, the tracker is reset upon a tracking failure [3]. The expected average overlap (*EAO*), a fundamental mix of accuracy and robustness, is used to assess performance (failure rate). The state-of-the-art trackers D3S [125], DSANet [133], DiMP-50 [118], KYS [130], SiamBAN [131], SiamRCNN [93], DiMP-18 [118], Ocean [134], ATOM [127], CCOT [34], ECO [10], SRDCF [10] are compared to CAM-DOT on the VOT dataset. The outcomes are shown in Table 7.1. In every metric, CAM-DOT performs better than all state-of-the-art trackers. On the VOT datasets, D3S [125] has recently been among the top trackers. The proposed tracking methodology derives its segmentation network from D3S. However, when co-attention is included in the tracking framework, the CAM-DOT tracker outperforms D3S in *EAO* by 20.89%, demonstrating the advantages of co-attention maps in object tracking.

Trackers	CAM-DOT	D3S	DSA Net	KYS	Siam BAN	DiMP -50	DiMP -18	Siam RCNN	ATOM	Ocean	ECO	CCOT	SRDCF
EAO	0.5909	0.4888	0.4698	0.4512	0.447	0.4468	0.4266	0.4044	0.4005	0.3849	0.2805	0.2671	0.1189

Table 7.1 VOT - state of the art comparison

7.3.3 Attribute based Evaluation

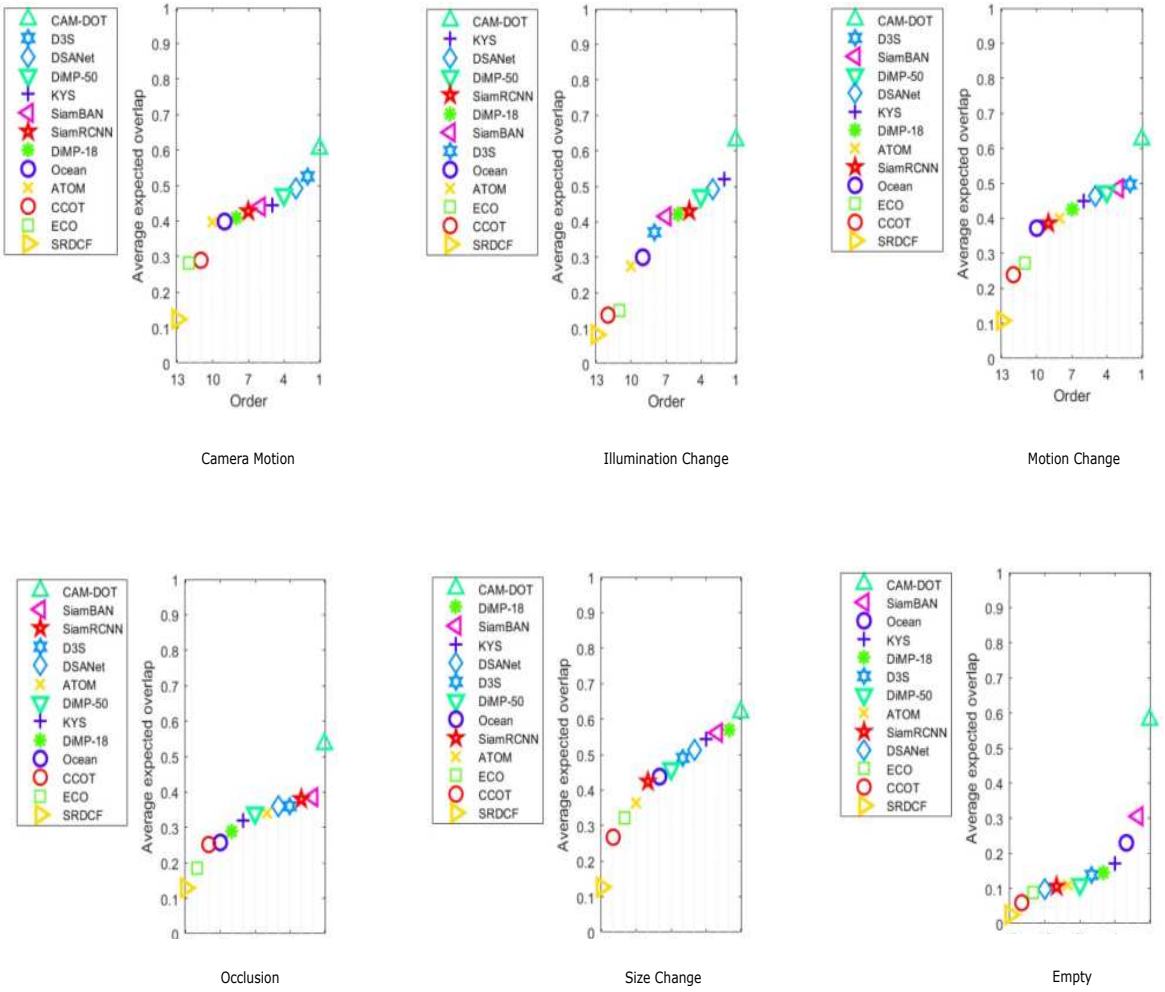


Fig. 7.6 Category wise evaluation of CAM-DOT on VOT dataset under various challenges and its state-of-the-art comparison.

The CAM-DOT tracker is evaluated attribute-wise under several conditions, including Camera Motion (CM), Illumination Change (IC), Motion Change (MC), Occlusion (OCC), Size Change (SC), and empty, which refers to frames without specific labels. Figure 7.6 displays the outcome of the attribute-wise evaluation. In all of the scenarios above, the CAM-DOT tracker performs better than the state-of-the-art trackers, which is attributable to the integration of co-attention into the correlation filter tracking framework.

	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
CAM-DOT	**0.927**	**0.919**	**0.901**	**0.964**	**0.946**	**0.932**	**0.874**	**0.903**	**0.948**	**0.939**	**0.959**
Siam-BAN	**0.915**	**0.919**	**0.874**	**0.956**	**0.917**	**0.948**	**0.799**	0.861	**0.946**	0.833	**0.93**
KYS	**0.907**	**0.928**	**0.871**	**0.959**	**0.924**	**0.926**	0.746	**0.893**	**0.942**	**0.892**	0.913
Siam-RCNN	0.904	0.872	0.861	0.85	0.88	0.871	**0.868**	0.868	0.892	**0.905**	**0.939**
DiMP-50	0.886	**0.891**	0.835	0.928	0.907	0.915	0.785	**0.87**	0.917	0.842	0.912
D3S	0.855	0.83	0.798	0.825	0.869	0.856	0.721	0.815	0.889	0.737	0.899
DiMP-18	0.846	0.829	0.774	0.805	0.828	0.807	0.613	0.826	0.842	0.768	0.867
DaSiam RPN	0.815	0.843	0.768	0.917	0.843	0.878	0.701	0.816	0.886	0.738	0.847
ATOM	0.803	0.789	0.714	0.851	0.823	0.826	0.691	0.842	0.849	0.747	0.88

Table 7.2 Attribute level precision comparison of proposed CAM-DOT tracker on OTB100. The compared attributes are fast motion (FM), background clutter (BC), motion blur (MB), deformation (DEF), Illumination Variation (IV), In-Plane-Rotation (IPR), Low Resolution (LR), occlusion (OCC), out-of-plane rotations (OPR), out-of-view (OV) and Scale Variations (SV). The top three trackers under each category are shown column wise in red, green, and blue respectively.

The proposed trackers are evaluated against a variety of tracking challenges, including fast motion (FM), background clutter (BC), motion blur (MB), deformation (DEF), illumination variation (IV), in-plane rotation (IPR), low resolution (LR), out-of-plane rotations (OPR), scale variations (SV), occlusions (OCC), and out-of-view (OV). This category analysis reveals the co-attention-based correlation filter tracking method's efficacy in attaining precise and long-term tracking. The top three ranks are indicated by the colors red, green, and blue, respectively, in Tables 7.2, 7.3, 7.4, which summarise the attribute level comparison on the OTB100 in terms of precision, Success (AUC), and overlap. The proposed CAM-DOT tracker performs exceptionally well under all of these challenges.

7.4 Chapter Summary

This chapter introduces a novel, single object tracking framework to improve the traditional correlation filter trackers. One of the critical elements of the suggested strategy is the use of the co-attention technique for visual object tracking. Both qualitative and quantitative assessments show that accurate target estimation and a lower failure rate illustrate the usefulness of the co-attention mechanism in tracking. The proposed tracker showed considerable

	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
CAM-DOT	**0.71**	**0.685**	**0.677**	**0.707**	**0.706**	**0.707**	**0.662**	**0.695**	**0.71**	**0.752**	**0.749**
Siam-BAN	**0.692**	**0.7**	**0.643**	**0.704**	**0.691**	**0.721**	**0.563**	0.652	**0.714**	0.668	**0.718**
KYS	**0.701**	**0.707**	**0.667**	**0.728**	**0.689**	**0.704**	0.574	**0.689**	**0.714**	**0.702**	**0.718**
Siam-RCNN	**0.71**	**0.686**	**0.661**	0.648	0.684	0.702	**0.675**	0.663	**0.692**	**0.736**	**0.742**
DiMP-50	0.682	**0.656**	0.638	0.697	0.665	0.678	**0.588**	**0.666**	0.683	0.673	0.706
D3S	0.635	0.597	0.577	0.597	0.636	0.625	0.521	0.583	0.638	0.588	0.658
DiMP-18	0.648	0.628	0.588	0.63	0.621	0.616	0.485	0.636	0.639	0.614	0.675
DaSiam RPN	0.602	0.627	0.561	0.669	0.62	0.65	0.49	0.611	0.653	0.591	0.63
ATOM	0.614	0.594	0.544	0.646	0.604	0.623	0.554	0.645	0.635	0.596	0.685

Table 7.3 Attribute level success (AUC) comparison of proposed CAM-DOT tracker on OTB100. The compared attributes are fast motion (FM), background clutter (BC), motion blur (MB), deformation (DEF), Illumination Variation (IV), In-Plane-Rotation (IPR), Low Resolution (LR), occlusion (OCC), out-of-plane rotations (OPR), out-of-view (OV) and Scale Variations (SV). The top three trackers under each category are shown column wise in red, green, and blue respectively

improvements in various challenging conditions. Future integration with more intricate tracking strategies will be more straightforward because of the generic nature of the proposed method.

	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
CAM-DOT	**0.908**	**0.887**	**0.879**	**0.916**	**0.915**	**0.898**	**0.844**	**0.89**	**0.91**	**0.931**	**0.942**
Siam-BAN	**0.873**	**0.895**	**0.806**	**0.925**	**0.893**	**0.914**	**0.707**	**0.851**	**0.922**	0.824	**0.904**
KYS	**0.882**	**0.91**	**0.847**	**0.925**	**0.889**	**0.895**	0.72	**0.865**	**0.909**	**0.862**	**0.897**
Siam-RCNN	**0.874**	**0.827**	**0.821**	0.789	0.834	0.857	**0.826**	0.83	**0.852**	**0.902**	**0.918**
DiMP-50	0.866	**0.847**	0.818	0.88	0.862	0.868	**0.752**	**0.838**	0.871	0.838	0.89
D3S	0.761	0.732	0.654	0.754	0.794	0.779	0.589	0.736	0.813	0.72	0.861
DiMP-18	0.813	0.793	0.749	0.797	0.785	0.76	0.589	0.806	0.804	0.753	0.852
DaSiam RPN	0.801	0.837	0.751	**0.906**	0.827	0.865	0.702	0.818	0.871	0.752	0.826
ATOM	0.771	0.763	0.688	0.828	0.783	0.776	0.67	0.819	0.808	0.74	0.869

Table 7.4 Attribute level overlap comparison of proposed CAM-DOT tracker on OTB100. The compared attributes are fast motion (FM), background clutter (BC), motion blur (MB), deformation (DEF), Illumination Variation (IV), In-Plane-Rotation (IPR), Low Resolution (LR), occlusion (OCC), out-of-plane rotations (OPR), out-of-view (OV) and Scale Variations (SV). The top three trackers under each category are shown column wise in red, green, and blue respectively

Chapter 8

Conclusion

Visual object tracking aims to determine the location of an object in subsequent video frames, given the initial state (centre location and scale) of the target in the first frame of a video sequence. An effective tracking algorithm should reliably track the object across time despite the appearance variations and environmental changes. Recently, the correlation filters have become highly popular for single object tracking. This acceptance of the correlation filter trackers is due to their computational simplicity, efficiency and robustness compared to other existing tracking algorithms.

Despite these advantages, the correlation filter trackers fail under several tracking scenarios. After careful study of the existing categories of correlation filter trackers this thesis has identified some generic drawbacks of the correlation filter trackers with respect to orientation changes, model drift and tracking resumption. Different techniques have been developed to address each of these problems.

The thesis initially considered the appearance variations due to object rotations. Orientation correction and illumination invariance were incorporated into the correlation tracker architecture. Further research developed three generic strategies to improve the precision and robustness of correlation filter trackers. The proposed techniques generate more reliable search regions and incorporate target-specific attention into correlation filter trackers, allowing for improved tracking, better resumption, and lower failure rates. The proposed trackers offer competitive performance in comparison to the most recent state-of-the-art trackers and significantly outperform the baseline algorithms in terms of accuracy and robustness.

Most correlation filter-based tracking techniques are sensitive to object deformations, rotations, and changes in illumination despite all the advancements in CF tracking. The appearance model is constrained by the lack of robust training features derived from each frame. In order to extract better features from each frame and aid in the development of a robust appearance model, the proposed work initially investigated the effects of oriented

bounding boxes over axis-aligned bounding boxes in each frame. The correlation filter is learned by optimization across oriented training samples. This helps in the extraction of robust features from correctly oriented bounding boxes. Further, the target movement is smoothened by considering the displacement across frames and false positives are eliminated by a new localization criterion based on correlation scores and distance from the previous location. The proposed tracking algorithm, discussed in Chapter 4, achieved significant improvements over the base trackers on benchmark datasets proving the efficiency of the approach in handling object rotations and deformations.

Tracking resumption was further identified as another limiting factor in correlation filter trackers. In Chapter 5, the thesis investigated the necessity for a generic technique to recognise tracking failures in correlation filter trackers. An online learned detector was then developed using Bag of Visual Words (BoVW). This detector re-initializes the tracker upon a target loss. The final target location was estimated using a similarity matching technique of the tracker and detector outputs with the initial target patch. This detector-based re-initialization technique has also achieved superior performance, especially in reducing the failure rate, compared to the base tracker.

Subsequently it was observed that the recursive search across the previous frame in conventional trackers results in tracking failures. This in turn generates an incorrect appearance model. The thesis further identified the need of confident search regions for target localization and developed a segmentation guided approach for single object tracking, as discussed in Chapter 6. The proposed method carefully incorporates domain-specific knowledge into the tracking pipeline via an initial offline training. Experimental analysis, both qualitative and quantitative, demonstrates the effectiveness of segmentation guided attention mechanism in tracking through accurate target estimation and failure rate reduction.

The thesis finally investigates the limited localization capability due to model drift and erroneous model update. The proposed solution is to find the common objects in adjacent frames. Inspired from the visual attention of human eyes, coattention maps that highlight the location of common objects in each search region is employed in the correlation filter tracker. In view of the better localization properties of segmented object masks, the research finally generates both segmented object masks and bounding boxes as target representations as detailed in Chapter 7.

These proposed approaches are generic and can be considered for integration with any correlation-based trackers. The research has identified the limiting conditions that are to be further investigated in the tracking scenario. The tracking algorithms demand improved accuracy in cases of very low resolutions, higher background clutter etc., together with real time performance. The benchmark datasets have started replacing the bounding box

representations with segmented object masks which marks another direction of research. Further these single object tracking algorithms can serve as the base for multiple object tracking, when integrated with suitable techniques for target associations. We hope that these works would serve as a guide to research in correlation based single object tracking and could be extended to real time societal applications like surveillance and monitoring systems.

Chapter 9

List of publications

Journals

- **Priya Mariam Raju**, Deepak Mishra and Rama Krishna Sai S. Gorthi. Detection based long term tracking in correlation filter trackers. *Pattern Recognition Letters*, Vol. 122, pp.79 - 85, 2019.
(https://doi.org/10.1016/j.patrec.2019.02.028)

- **Priya Mariam Raju**, Deepak Mishra and Prerana Mukherjee. DA-SACOT: Domain adaptive-segmentation guided attention for correlation based object tracking. *Image and Vision Computing* Vol. 112, 2021.
(https://doi.org/10.1016/j.imavis.2021.104215)

- **Priya Mariam Raju**, and Deepak Mishra, CAM-DOT: Co-Attention Maps for Discriminative Object Tracking. (under preparation)

Conference Proceedings

- **Priya Mariam Raju**, Deepak Mishra and Rama Krishna Sai S Gorthi, Visual Object Challenge Results 2018, *In Proceedings of the European Conference on Computer Vision (ECCV)*, Visual object Tracking Workshop, 2018.

- **Priya Mariam Raju**, Deepak Mishra and Rama Krishna Sai S Gorthi, Bag of Visual Words based Correlation Filter Tracker, *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, December, 2018.

- Litu Rout, **Priya Mariam Raju**, Deepak Mishra, and Rama Krishna Sai Subrahmanyam Gorthi, Learning Rotation Adaptive Correlation Filters in Robust Visual Object Tracking, *In Asian Conference on Computer Vision*, 2018.

References

[1] M. Kristan *et al.*, "A novel performance evaluation methodology for single-target trackers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 2137–2155, Nov 2016.

[2] M. Danelljan *et al.*, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4310–4318, 2015.

[3] M. Kristan *et al.*, "The seventh visual object tracking vot2019 challenge results," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

[4] A. Ali *et al.*, "Visual object tracking—classical and contemporary approaches," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 167–188, 2016.

[5] B. Tian *et al.*, "Video processing techniques for traffic flow monitoring: A survey," in *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems*, pp. 1103–1108, 2011.

[6] S. Walker *et al.*, "Systems and methods for localizing, tracking and/or controlling medical instruments," Apr. 25 2017. US Patent 9,629,595.

[7] J. K. Aggarwal and L. Xia, "Human activity recognition from 3d data: A review," *Pattern Recognition Letters*, vol. 48, pp. 70–80, 2014.

[8] V. A. Laurense, J. Y. Goh, and J. C. Gerdes, "Path-tracking for autonomous vehicles at the limit of friction," in *Proceedings of the IEEE American Control Conference (ACC)*, pp. 5586–5591, 2017.

[9] S. Joan, "Human-digital media interaction tracking," *US Patent*, no. 9, p. 713, 2017.

[10] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6638–6646, 2017.

[11] M. Danelljan *et al.*, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1090–1097, 2014.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[13] K. He *et al.*, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[14] L. Bertinetto *et al.*, "Staple: Complementary learners for real-time tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1401–1409, 2016.

[15] Z. Chi *et al.*, "Dual deep network for visual tracking," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 2005–2015, 2017.

[16] M. Danelljan *et al.*, "Discriminative scale space tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2016.

[17] H. Fan and H. Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5486–5494, 2017.

[18] Z. Hong *et al.*, "Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 749–758, 2015.

[19] C. Ma *et al.*, "Robust visual tracking via hierarchical convolutional features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2709–2723, 2018.

[20] C. Ma *et al.*, "Long-term correlation tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5388–5396, 2015.

[21] Y. Qi *et al.*, "Hedged deep tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4303–4311, 2016.

[22] Y. Song *et al.*, "Crest: Convolutional residual learning for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2555–2564, 2017.

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

[24] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4021–4029, 2017.

[25] M. Zhang *et al.*, "Robust visual tracking using joint scale-spatial correlation filters," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 1468–1472, 2015.

[26] M. Zhang *et al.*, "Joint scale-spatial correlation tracking with adaptive rotation estimation," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 32–40, 2015.

[27] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4335–4343, 2017.

[28] C. Ma *et al.*, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3074–3082, 2015.

[29] Y. Qi *et al.*, "Hedged deep tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4303–4311, 2016.

[30] M. Ozuysal, P. Fua, and V. Lepetit, "Fast keypoint recognition in ten lines of code," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[31] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4335–4343, 2017.

[32] J. Choi *et al.*, "Attentional correlation filter network for adaptive visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4807–4816, 2017.

[33] J. Choi *et al.*, "Visual tracking using attention-modulated disintegration and integration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4321–4330, 2016.

[34] M. Danelljan *et al.*, "Convolutional features for correlation filter based visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 58–66, 2015.

[35] M. Danelljan *et al.*, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1430–1438, 2016.

[36] M. Danelljan *et al.*, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 472–488, 2016.

[37] S. Gladh *et al.*, "Deep motion features for visual tracking," in *Proceedings of the 23rd IEEE International Conference on Pattern Recognition (ICPR)*, pp. 1243–1248, 2016.

[38] H. Hu *et al.*, "Manifold regularized correlation object tracking," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1786–1795, 2017.

[39] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1135–1143, 2017.

[40] F. Li *et al.*, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4904–4913, 2018.

[41] A. Lukezic *et al.*, "Discriminative correlation filter with channel and spatial reliability," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6309–6318, 2017.

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

[43] L. Bertinetto *et al.*, "Fully-convolutional siamese networks for object tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 850–865, 2016.

[44] Q. Guo *et al.*, "Learning dynamic siamese network for visual object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1763–1771, 2017.

[45] C. Huang, S. Lucey, and D. Ramanan, "Learning policies for adaptive tracking with deep feature cascades," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 105–114, 2017.

[46] J. Valmadre *et al.*, "End-to-end representation learning for correlation filter based tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2805–2813, 2017.

[47] Q. Wang *et al.*, "Dcfnet: Discriminant correlation filters network for visual tracking," *arXiv preprint arXiv:1704.04057*, 2017.

[48] K. Chen, W. Tao, and S. Han, "Visual object tracking via enhanced structural correlation filter," *Information Sciences*, vol. 394, pp. 232–245, 2017.

[49] W. Chen, K. Zhang, and Q. Liu, "Robust visual tracking via patch based kernel correlation filters with adaptive multiple feature ensemble," *Neurocomputing*, vol. 214, pp. 607–617, 2016.

[50] Z. Cui *et al.*, "Recurrently target-attending tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1449–1458, 2016.

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

[52] S. Liu *et al.*, "Structural correlation filter for robust visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4312–4320, 2016.

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

[54] A. Lukežič, L. Č. Zajc, and M. Kristan, "Deformable parts correlation filters for robust visual tracking," *IEEE Transactions on Cybernetics*, vol. 48, no. 6, pp. 1849–1861, 2017.

[55] B. Bai *et al.*, "Kernel correlation filters for visual tracking with adaptive fusion of heterogeneous cues," *Neurocomputing*, vol. 286, pp. 109–120, 2018.

[56] C. Luo *et al.*, "Comparison of different level fusion schemes for infrared-visible object tracking: An experimental survey," in *2nd IEEE International Conference on Robotics and Automation Sciences (ICRAS)*, pp. 1–5, 2018.

[57] L. Ma *et al.*, "Multiple feature fusion via weighted entropy for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3128–3136, 2015.

[58] M. K. Rapuru *et al.*, "Correlation-based tracker-level fusion for robust visual tracking," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4832–4842, 2017.

[59] G. Wang *et al.*, "Robust visual tracking with deep feature fusion," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1917–1921, 2017.

[60] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2011.

[61] J. F. Henriques *et al.*, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 583–596, 2014.

[62] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[63] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 445–461, Springer, 2016.

[64] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5630–5644, 2015.

[65] D. Martin *et al.*, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the 8th International Conference on Computer Vision*, vol. 2, pp. 416–423, July 2001.

[66] P. J. Phillips *et al.*, "The feret evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[67] H. Shi *et al.*, "Panoflow: Learning 360 optical flow for surrounding temporal understanding," *arXiv preprint arXiv:2202.13388*, 2022.

[68] A. Aghamohammadi *et al.*, "A parallel spatio-temporal saliency and discriminative online learning method for visual target tracking in aerial videos," *PloS one*, vol. 13, no. 2, 2018.

[69] J. L. Crowley, P. Reignier, and S. Pesnel, "Context aware vision using image-based active recognition," 2004.

[70] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[71] S. Hadfield, K. Lebeda, and R. Bowden, "The visual object tracking vot2014 challenge results," in *Proceedings of the European Conference on Computer Vision (ECCV) Visual Object Tracking Challenge Workshop*, 2014.

[72] M. Kristan *et al.*, "The visual object tracking vot2015 challenge results," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1–23, 2015.

[73] A. W. Smeulders *et al.*, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2013.

[74] D. S. Bolme *et al.*, "Visual object tracking using adaptive correlation filters," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, 2010.

[75] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, no. 1, pp. 99–110, 2003.

[76] S. Oron *et al.*, "Locally orderless tracking," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 213–228, 2015.

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

[78] H. Nam, M. Baek, and B. Han, "Modeling and propagating cnns in a tree structure for visual tracking," *arXiv preprint arXiv:1608.07242*, 2016.

[79] L. Rout *et al.*, "Rotation adaptive visual object tracking with motion consistency," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1047–1055, 2018.

[80] Q. Du *et al.*, "A rotation adaptive correlation filter for robust tracking," in *Proceedings of the IEEE International Conference on Digital Signal Processing*, pp. 1035–1038, 2015.

[81] M. M. Petrou and C. Petrou, *Image processing: the fundamentals*. John Wiley & Sons, 2010.

[82] S. Hadfield, R. Bowden, and K. Lebeda, "The visual object tracking vot2016 challenge results," *Lecture Notes in Computer Science*, vol. 9914, pp. 777–823, 2016.

[83] C. Sun *et al.*, "Learning spatial-aware regressions for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8962–8970, 2018.

[84] Z. Zhu *et al.*, "End-to-end flow correlation tracking with spatial-temporal attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 548–557, 2018.

[85] C. Ma *et al.*, "Long-term correlation tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5388–5396, 2015.

[86] Y. Wang *et al.*, "Effective multi-query expansions: Collaborative deep networks for robust landmark retrieval," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1393–1404, 2017.

[87] G. Ning *et al.*, "Spatially supervised recurrent convolutional neural networks for visual object tracking," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, 2017.

[88] M. Kristan *et al.*, "The eighth visual object tracking vot2020 challenge results," in *Proceedings of the European Conference on Computer Vision Workshops*, pp. 547–601, Springer, 2020.

[89] K. He *et al.*, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, 2017.

[90] A. Zgaren, W. Bouachir, and R. Ksantini, "Coarse-to-fine object tracking using deep features and correlation filters," in *International Symposium on Visual Computing*, pp. 517–529, Springer, 2020.

[91] N. Wang *et al.*, "Multi-cue correlation filters for robust visual tracking," in *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition*, pp. 4844–4853, 2018.

[92] K. Dai *et al.*, "Visual tracking via adaptive spatially-regularized correlation filters," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4670–4679, 2019.

[93] P. Voigtlaender *et al.*, "Siam r-cnn: Visual tracking by re-detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6578–6588, 2020.

[94] Y. Wang *et al.*, "Detection based visual tracking with convolutional neural network," *Knowledge-Based Systems*, vol. 175, pp. 62–71, 2019.

[95] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3038–3046, 2017.

[96] B. Yang *et al.*, "A vehicle tracking algorithm combining detector and tracker," *EURASIP Journal on Image and Video Processing*, vol. 2020, no. 1, pp. 1–20, 2020.

[97] R. Yao *et al.*, "Video object segmentation and tracking: A survey," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 4, pp. 1–47, 2020.

[98] Z. Cai *et al.*, "Robust deformable and occluded object tracking with dynamic graph," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5497–5509, 2014.

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

[100] M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," *Computer Vision and Image Understanding*, vol. 117, no. 10, pp. 1245–1256, 2013.

[101] S. Wang *et al.*, "Superpixel tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1323–1330, 2011.

[102] V. Belagiannis *et al.*, "Segmentation based particle filtering for real-time 2d object tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 842–855, Springer, 2012.

[103] M. Keuper *et al.*, "Motion segmentation & multiple object tracking by correlation co-clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 1, pp. 140–153, 2018.

[104] R. Yao *et al.*, "Semantics-aware visual object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1687–1700, 2018.

[105] D. Yeo *et al.*, "Superpixel-based tracking-by-segmentation using markov chains," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1812–1821, 2017.

[106] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5188–5197, 2019.

[107] G. Bertasius and L. Torresani, "Classifying, segmenting, and tracking object instances in video with mask propagation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9739–9748, 2020.

[108] P. Voigtlaender *et al.*, "Mots: Multi-object tracking and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7942–7951, 2019.

[109] K. He *et al.*, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[110] J. Deng *et al.*, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, IEEE, 2009.

[111] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision*, pp. 740–755, Springer, 2014.

[112] N. Wang *et al.*, "Unsupervised deep tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1308–1317, 2019.

[113] T. Yang *et al.*, "Roam: Recurrently optimizing tracking model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6718–6727, 2020.

[114] Z. Chen *et al.*, "Siamese box adaptive network for visual tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6668–6677, 2020.

[115] X. Li *et al.*, "Target-aware deep tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1369–1378, 2019.

[116] J. Gao, T. Zhang, and C. Xu, "Graph convolutional tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4649–4659, 2019.

[117] B. Li *et al.*, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4282–4291, 2019.

[118] G. Bhat *et al.*, "Learning discriminative model prediction for tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6182–6191, 2019.

[119] Z. Zhu *et al.*, "Distractor-aware siamese networks for visual object tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 101–117, 2018.

[120] M. Danelljan, L. V. Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7183–7192, 2020.

[121] G. Bhat *et al.*, "Unveiling the power of deep tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 483–498, 2018.

[122] S. Hare *et al.*, "Struck: Structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096–2109, 2015.

[123] M. Kristan *et al.*, "The sixth visual object tracking vot2018 challenge results," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

[124] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2113–2120, 2015.

[125] A. Lukezic, J. Matas, and M. Kristan, "D3s-a discriminative single shot segmentation tracker," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7133–7142, 2020.

[126] W. Li, O. Hosseini Jafari, and C. Rother, "Deep object co-segmentation," in *Proceedings of the Asian Conference on Computer Vision*, pp. 638–653, Springer, 2018.

[127] M. Danelljan *et al.*, "Atom: Accurate tracking by overlap maximization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4660–4669, 2019.

[128] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[129] M. Everingham *et al.*, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98–136, Jan. 2015.

[130] G. Bhat *et al.*, "Know your surroundings: Exploiting scene information for object tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 205–221, Springer, 2020.

[131] Z. Chen *et al.*, "Siamese box adaptive network for visual tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6668–6677, 2020.

[132] Z. Zhu *et al.*, "Distractor-aware siamese networks for visual object tracking," in *European Conference on Computer Vision*, 2018.

[133] M. A. Elhassan *et al.*, "Dsanet: Dilated spatial attention for real-time semantic segmentation in urban street scenes," *Expert Systems with Applications*, 2021.

[134] Z. Zhang *et al.*, "Ocean: Object-aware anchor-free tracking," in *European Conference on Computer Vision*, pp. 771–787, Springer, 2020.