Multi Cue Robust Visual Object Tracking

Baber Khan

78-FET/PHDEE/F14

:

Submitted in partial fulfillment of the requirements for the PhD degree in Electronic Engineering at the Department of Electrical Engineering

Faculty of Engineering and Technology

International Islamic University,

Islamabad

Supervisor	Co Supervisor
Prof. Dr. Abdul Jalil	Dr. Ahmad Ali

August, 2021

Sensor Network
Electronic security systems
Signal processing. Digital

Multi Cue Robust Visual Object Tracking

Researcher:

Baber Khan

REG NO. 78-FET/PHDEE/F14

Supervisor:

Prof. Dr. Abdul Jalil

Co Supervisor:

Dr. Ahmad Ali

Department of Electrical Engineering

Faculty of Engineering & Technology

INTERNATIONAL ISLAMIC UNIVERSITY,

ISLAMABAD

CERTIFICATE OF APPROVAL

Title of Thesis: Multi Cue Robust Visual Object Tracking

Name of Student: Baber Khan

Registration No: 78-FET/PhDEE/F14

Accepted by the Department of Electrical Engineering, Faculty of Engineering and Technology, International Islamic University, Islamabad, in partial fulfillment of the requirements for the Doctor of Philosophy degree in Electronic Engineering.

Viva voce committee:

Dr. Ahmad Ali (Co-Supervisor)
General Manager, NESCOM, Islamabad

Dr. Abdul Jalil (Supervisor)
Ex-Professor, Department of Electrical Engineering
International Islamic University, Islamabad

Prof. Dr. Aqdas Naveed Malik (Internal Examiner)
Professor, Department of Electrical Engineering
International Islamic University, Islamabad.

Dr. Rab Nawaz Khan (External Examiner - I)
Project Director, NESCOM, Islamabad

Dr. Noaman Ahmad Khan (External Examiner - II)
Professor/Chairman, DEE, Sir Syed CASE Institute of
Technology B-17, Islamabad.

Dr. Suheel Abdullah Malik (Chairman)
Associate Professor, Department of Electrical Engineering
International Islamic University, Islamabad.

Prof. Dr. Nadeem Ahmed Sheikh (Dean)
Professor, Department of Mechanical Engineering,
Faculty of Engineering & Technology,
International Islamic University, Islamabad.

February 17, 2022

DEDICATED TO

My teachers,

Parents,

Wife,

Brothers,

Sisters,

Sons (M. Abdullah & Ahmed Sadeed)

ABSTRACT

During recent years correlation tracking has been considered fast and effective by virtue of the circulant structure of the sampling data for the learning phase of filter and Fourier domain calculation of correlation. During occlusion, motion blur, and out-of-view movement of a target, most of the correlation filter-based trackers start to learn using erroneous samples and the tracker starts drifting. Currently, adaptive correlation filter-based tracking algorithms are being combined with redetection modules. This hybridization helps in redetection of the target in long-term tracking. The redetection modules are mostly classifiers, which classify the true object after tracking failure occurrence. These methods perform favorably during short-term occlusion or partial occlusion. To further increase the tracking efficiency in challenging video sequences, specifically during long-term occlusion, while maintaining real-time processing speed, this study presents a tracking failure avoidance method, efficient occlusion detection, and handling mechanism, and a new adaptive learning rate strategy. We first present a strategy to detect the occlusion using multiple cues from the response map, i.e., peak correlation score and peak to side lobe ratio, average peak correlation energy, the confidence of squared response map. We further introduce a novel interpretation of the difference of peak correlation between two consecutive frames. After successful detection of tracking failure using multiple cues, a second strategy is presented to save the target being getting more erroneous. Our predictor in the prediction-estimation collaboration module continuously predicts the location during occlusion. The predictor passes this result to Support Vector Machine (SVM). When the target reappears in a frame, the support vector machine-based classifier finds the correct object using the predicted location. This collaboration between prediction and estimation decreases the chance of tracking failure as the predictor continuously updates itself during occlusion and predicts the next location using its previous prediction. Once the true object is detected by the classifier after the clearance of occlusion, this result is forwarded to the baseline tracker to resume its tracking operation and update its parameters. Together these two proposed schemes show significant improvement in tracking efficiency. Furthermore,

this collaboration in redetection phase shows significant improvement in the tracking accuracy over videos containing six challenging aspects of visual object tracking as mentioned in the literature. Novel adaptive learning rate strategy further increases the robustness of proposed scheme. Comparison with 14 state-of-the-art algorithms is given in this study. For evaluation of results, three different standard datasets are used. This comparison shows that outcome of this study performs better than the other 14 state-of-the-art algorithms.

LIST OF PUBLICATIONS AND SUBMISSIONS

[1]. **B. Khan, A. Ali, A. Jalil, K. Mehmood. M. Murad and H. Awan, "AFAM-PEC: Adaptive Failure Avoidance Tracking Mechanism Using Prediction-Estimation Collaboration," in *IEEE Access*, vol. 8, pp. 149077-149092, 2020, doi: 10.1109/ACCESS.2020.3015580.**

[2]. **B. Khan, A Jalil, A. Ali, K. Mehmood, M. Murad, " Multi cue based robust visual object tracking" submitted. in Electronics. Vol 11, 2022.**

[3]. **K. Mehmood, A. Jalil, A. Ali, B. Khan, KM Cheema, M. Murad, AH. Milyani, "Efficient Online Object Tracking Scheme for Challenging Scenarios" Sensors 2021, 21, 8481**

[4]. **K. Mehmood, A. Jalil, A. Ali, B. Khan, M. Murad, W. Khan, H. Yigang " Context-Aware and Occlusion Handling Mechanism for Online Visual Object Tracking" in Electronics, Vol 10, 2021.**

[5]. **K. Mehmood, A. Jalil, A. Ali, B. Khan, M. Murad, KM Cheema, AH. Milyani "Spatio-Temporal Context, Correlation Filter and Measurement Estimation Collaboration Based Visual Object Tracking" in Sensors, Vol 21, pages 2841, 2021.**

[6]. **M. Murad, A. Jalil, M. Bilal, S. Ikram, A. Ali, K. Mehmood, B. Khan "Gaussian-Radial Under-Sampling Based CSMRI Reconstruction using a Modified Interpolation Approach," 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), 2021, pp. 1-6.**

[7]. **M. Murad, M. Bilal, A. Jalil, A. Ali, K. Mehmood, B. Khan "Efficient reconstruction technique for multi-slice CS-MRI using novel interpolation and 2D sampling scheme" in IEEE Access, Vol 8, pages 117452-117466, 2020/6/24.**

[8]. **M. Murad, A. Jalil, M. Bilal, S. Ikram, A. Ali, B. Khan, K. Mehmood "Radial Undersampling-Based Interpolation Scheme for Multislice CSMRI Reconstruction Techniques" in BioMed Research International, Vol 2021, 2021/4/12.**

[9]. **MN. Tajik, AU. Rehman, W. Khan, B. Khan "Texture Feature Selection Using GA for Classification of Human Brain MRI Scans" in International Conference in Swarm Intelligence, Bali, Indonesia, 6/2016.**

[10]. **F. Khan, AR. Rehman, M. Arif, B. Khan, M. Aftab "A survey of communication technologies for smart grid connectivity" in 2016 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube), 4/2016.**

[11]. AU.Rehman, F. Khan, B. Khan, "Analysis of adaptive filter and ICA for noise cancellation from a video frame" in 2016 International Conference on Intelligent Systems Engineering (ICISE), 1/2016.

[12]. B. Khan, AK. Khan, G. Raja, MH. Yousaf "Implementation of Modified Mean-shift Tracking Algorithm for Occlusion Handling" in Life Science Journal 10(11s):337-342.

The following study is based on the first two articles.

ACKNOWLEDGEMENTS

In the name of Allah (Subhanahu Wa Ta'ala), who is the most Gracious and the Merciful. I would like to thank Allah for giving me the strength and patience to complete this research work. Peace and blessings of Allah be upon His last Prophet Muhammad (Sallulah-o-Alaihihe-Wassalam) and all his Sahaba (Razi-Allah-o-Anhu) who dedicated their lives for Dawah and spread of Knowledge.

I am truly grateful to my supervisor Dr. Abdul Jalil whose inspiration, ideas and efforts make it possible for me to complete my higher studies. He has been a role model for me and many others in teaching, research, and other aspects of life. I would also like to thank my mentor and co-supervisor, Dr. Ahmed, who always helped me understand things deeply.

I offer my sincere thanks to my colleagues and friends including Engr. Khizer, Dr. Naveed Ishtiaq Chaudhry, Dr. Athar Waseem, Dr. Zeeshan Aslam Dr. Atiq ur Rehman and Dr. Hamdan their never-ending support and fruitful and healthy research discussions. I would like to acknowledge the support of International Islamic University Islamabad, Pakistan for providing me a full fee waiver during the PhD studies. I am thankful to administration at the department, as well as the university level for their kind support.

I am grateful to my father, mother, brothers and sister for their love, good wishes, and support throughout my life. I am also very indebted to my wife for her patience, encouragement, and prayers during every single stage of my PhD degree.

(Baber Khan)

Table of Contents

ABSTRACT iii
LIST OF PUBLICATIONS AND SUBMISSIONS v
ACKNOWLEDGEMENTS vii
LIST OF FIGURES xi
LIST OF SYMBOLS xii
Chapter 1. 1
Introduction 1
1.1 Overview 1
1.2 Applications 3
1.2.1 Autonomous driving cars 3
1.2.2 Sports activity analysis 3
1.2.3 Unmanned Aerial vehicles 4
1.2.4 Human-machine interaction 4
1.2.5 Visual surveillance 4
1.2.6 Image based health diagnostic systems 5
1.3 Challenging issues associated with visual object tracking 5
1.3.1 Occlusion 5
1.3.2 Appearance change 5
1.3.3 Cluttered background 6
1.3.4 Scale variation 6
1.3.5 Complex object motion 6
1.3.6 Illumination variation 6
1.4 Background and Motivation 6
1.5 Scope and contributions 7
1.6 Research problem statement 9
1.7 Research objectives 10
1.8 Structure of thesis 10
Chapter 2. 12
Literature Review 12

2.1 Image recognition 12
2.2 The sampling problem 15
2.3 Fourier-domain methods 16
2.4 Tracking learning and detection (tracking by detection) 17
2.5 Correlation filter tracking 18
2.5.1 Minimum Output Sum of Squared Error Filter 20
2.5.2 Kernelized Correlation Filter 22
2.5.3 A Simple Correlation Filter Tracker 25
2.6 Improvements for Correlation Filter Trackers 28
2.6.1 Feature Representations for Correlation Filter Trackers 28
2.7 Target scale estimation 33
2.7.1 Exhaustive scale search 33
2.7.2 Efficient scale search 35
2.8 Parts-based correlation filter trackers 36
2.9 Other visual trackers 39
2.10 Summary 42
Chapter 3. ..... **43**
Methodology **43**
3.1 Long-term correlation tracking 43
3.2 Support vector machine-based estimator 46
3.3 Kalman filter-based prediction 50
3.4 Prediction estimation collaboration 51
3.5 Hybridization of average peak correlation energy and confidence of squared response map 53
3.6 Novel interpretation of difference of peak correlation 54
3.7 Implementation details 56
Chapter 4. ..... **58**
Results and Discussion **58**
4.1 Dataset 58
4.1.1 Attributes of dataset 62
4.2 Quantitative analysis 65

4.3 Qualitative analysis 74
4.4 Summary 82
Chapter 5. ..... **83**
Conclusion and Future Work **83**
5.1 Conclusion 83
5.2 Future recommendations 84
BIBILOGRAPHY **85**

LIST OF FIGURES

Figure 2.1	Standard correlation filter	36
Figure 2.2	Part-based correlation tracking	37
Figure 3.1	Support vector machine demonstration	45
Figure 3.2	Pictorial representation of the proposed algorithm (first paper)	48
Figure 3.3	Modeling of context regression model	49
Figure 3.4	Flow chart of prediction-estimation collaboration	52
Figure 3.5	Proposed graphical abstract of modified KCF algorithm (second paper)	55
Figure 4.1	Comparison of LSTM and Proposed algorithm for jogging1..	59
Figure 4.2	Comparison of LSTM and Proposed algorithm for jogging2..	60
Figure 4.3	Results without prediction-estimation collaboration	61
Figure 4.4	Qualitative results for.....	63
Figure 4.5	Qualitative results for.....	64
Figure 4.6	Overlap success rate comparison with LSTM	66
Figure 4.7	Overlap success rate comparison with STC	67
Figure 4.8	Overlap success rate comparison with CT	68
Figure 4.9	Overlap success rate comparison with LSTM, STC and CT	69
Figure 4.10	Peak correlation score of proposed work and LSTM.....	70

LIST OF SYMBOLS

A list of commonly used symbols in this dissertation are given below.

Denotation	Symbol	Note
Estimated position and scale	$x_t = (\hat{x}_t, \hat{y}_t, \hat{s}_t)$,	$\hat{x}_t$, $\hat{y}_t$, position of the object and $\hat{s}_t$ is estimated scale
Correlation response map	y_t	At t^{th} frame
Regression model	R_{con}	With respect to context
Regression model	R_{tar}	Model of target
Detector module	D_{svm}	SVM based
Predictor module	P_{kf}	Kalman based
Estimated new position	$\hat{y}_t$	At t^{th} frame
Predicted state	x_t''	State by P_{kf} at tth frame
Estimated states	$X(D_{rf})$	All possible states by detector
Estimated state	x'_i	Estimated possible state I for t^{th} frame
Response map value	y'_i	Response map value for estimated state i

Chapter 1.

Introduction

This chapter discusses the introduction of visual object tracking and the challenges associated with tracking. Furthermore, background along with motivation is provided. At last, brief thesis contributions and an overview of the thesis is provided.

1.1 Overview

Visual object tracking has always been considered an active area of interest in the research field of computer vision because of its wide spread applications and challenging issues like motion blur, object deformation, noisy environment, fast motion, clutter, and finally occlusion [1], [2]. Long-term tracking is considered effective if an algorithm tracks an object of interest for a long time in all or any of the above challenging scenarios. Without considering the orientation estimation of the object, the tracking process can be divided into two subparts i.e. i) translation estimation and ii) scale estimation of a target in the next frame [3].

For translation estimation, broadly tracking algorithms can be divided into two groups: i) generative and ii) discriminative. In the generative scheme, the information of the object is used while considering tracking as a search problem. The discriminating scheme considers the tracking a classification problem, while using the object and its background information. Discriminative tracking using a correlation filter is studied several researchers in the field of object tracking [4], [5], [6], [7], [8], [9], [10], [11]. Exploiting circular structure and computing

correlation in the frequency domain which is simply multiplication, the extreme fast tracker is presented in [12]. Due to the adaptive nature of the correlation filter, the online fast learning mechanism makes the correlation filter suitable for fast appearance-changing object tracking. Though correlation filters are very successful in visual object tracking but still two major limitations exist; First, they do not have the inherent capability of tracking resumption once the object is lost or the object moves out of the camera's field of view. The second is that the less reliable tracked frame causes the correlation filter to learn the wrong target appearance, and this learning error accumulates with the passage of frames. The first limitation is addressed in [9], [13] by considering the redetection module, where redetection is carried out in each frame, increasing the complexity cost. Another approach to reducing the computational cost is defining a threshold to activate the redetection module [3], [14]. The second limitation of correlation filter based tracking is solved in [3] by learning multiple correlation filters with different learning rates. To cover the fixed template size problem of kernel filters, correlation filter adaptive to scale changing is presented in [15]. Dense spatio-temporal context information is used in [16] to increase the efficiency and robustness of the correlation filters. A simple tracking approach with an appearance model based on multi-scale image feature extraction using a data-independent basis is presented in [17]. Particle filters are also incorporated in kernelized correlation filters to redetect the tracker when response map becomes less reliable [18]. Fusion of multiple features in the correlation filter framework is proposed in [19]. In this method adaptive weights are assigned to each feature to minimize the interference of noise. The metric learning model strategy is given in [20] to enhance the quality of response map in correlation filter-based algorithms. Numerous researcher also proposed

Convolutional Neural Network based-tracking strategies during recent years. For latest examples, see [21], [22]. These neural network-based algorithms require a lot of training data and large computational time.

1.2 Applications

Visual tracking has been used in applications like robotics, surveillance, sports analysis, unmanned aerial vehicle, image based health diagnostic systems, activity recognition, industrial robotics, lip tracking in film industry, transportation, and autonomous driving cars. Some of them are presented in this section.

1.2.1 Autonomous driving cars

Autonomous cars have been considered an important application of visual object tracking since the last decade. It is a vehicle without a driver that can sense the nearby environment to avoid any obstacle. These cars are equipped with many different systems like the navigation system, path planning, environment perception, and control systems. For environment perception, visual object tracking can be used to recognize nearby objects and get the position. This information extracted from video is helpful for a car to plan its path by avoiding other objects. Furthermore, information regarding the position of the objects at different time instants also helps the car to predict the future position of the surrounding object. Finally, visual object tracking can also locate the position of traffic signs for traffic sign recognition.

1.2.2 Sports activity analysis

During recent years, motion analysis of players has gained a lot of attention from the visual object tracking community. Without object tracking, activity is analyzed by observing the data

collected after the activity. Traditionally people manually record and analyze the data. With the help of visual object tracking, this process is automated. Let us take the example of a basketball game. Visual object tracking help in predicting the trajectories of players. Furthermore, this information is used for strategy planning and performance evaluation of players and team management.

1.2.3 Unmanned Aerial vehicles

The unmanned aerial vehicle has been used widely for surveillance, product deliveries, arial photography and inspections. In all these applications, camera mounted on unmanned aerial vehicle plays an essential role with the help of visual object tracking. For example, a user selects the target in the video frame, and unmanned aerial vehicle will process the video captured by a camera and find the target's position. With the help of this position, an unmanned aerial vehicle can adjust itself to follow the target constantly.

1.2.4 Human-machine interaction

The visual object tracking community plays a vital part in helping the community by providing efficient and user-friendly interaction with machines, for example, providing sixth sense to humans i.e., a wearable gesture interface, perceptual user interfaces, eye gaze tracking for visually impaired people etc.

1.2.5 Visual surveillance

Nowadays visual object tracking is an integral part of efficient and intelligent visual surveillance systems. Like, Siemens siemons sistore CX EDS-intelligent video detection

system, surveillance of open places, parks, colonies, and buildings for suspicious activity detection.

1.2.6 Image based health diagnostic systems

Visual object tracking methods are also being applied in health diagnostic system. For example, ventricular wall tracking, and reconstruction of vocal tract shape.

1.3 Challenging issues associated with visual object tracking

For the last 2-3 decades, visual object tracking has been considered an essential area for research because of challenges associated with it like occlusion, motion blur, out of view movement of the object, illumination changes in video, in-plane rotation, out of the plane rotation, clutter background, appearance changes, and object deformation.

1.3.1 Occlusion

Occlusion is considered a widespread and challenging problem for the object tracking community. When the target to be tracked is occluded by some other object, this phenomenon is called occlusion. It is further classified as partial occlusion and full occlusion. Strategies are designed by observing the environment and nature of the target.

1.3.2 Appearance change

Most of the time in tracking, the object to be tracked is non-rigid, which may change its appearance during motion. To handle this issue, mostly adaptive tracking schemes are applied. During the update of the target model, even small errors accumulate as time passes and drift problem arise. If the model is kept fixed, changes in target appearance would not be

incorporated and again target will be lost. This is known as stagnation to the old appearance problem. Hence there is a trade-off between drift and stagnation.

1.3.3 Cluttered background

Often in object tracking background is not a single object. When the background has many other objects, it is known as clutter in object tracking terminology. This problem is easily handled when the background is known but if the background is unknown just like in outdoor tracking, the severity of the problem increases.

1.3.4 Scale variation

Target changes its size when it moves away or toward the camera. Therefore, tracking schemes need to adjust the template/target model size accordingly.

1.3.5 Complex object motion

This includes out-of-plane movement or abrupt change in speed and direction of the target. Due to the wrong approximation of the target model, tracking becomes a more difficult task.

1.3.6 Illumination variation

Change in light is also one of the major challenges for the object tracking community. When the object is moving from dark to light or vice versa, it changes its appearance.

1.4 Background and Motivation

Visual object tracking is a secondary field of computer vision and machine learning. This field is interesting because of its usefulness in real-world applications for real -ime scenarios. One of the interesting applications is the use of visual object tracking in robotics. Tracking

algorithms give sight to a robot just like the human eye and track the objects as per requirement. Surveillance is also a fascinating example of visual object tracking. In the current world, closed-circuit television cameras are installed at every important place to record the activities. Most of the time, these cameras record the garbage. For example, there is no activity, but cameras are still recording, resulting in over usage of memory with garbage data. When these cameras are equipped with visual object tracking algorithms, only suspicious activities are saved in the memory. This helps in the efficient utilization of memory. One other aspect tracking and saving only the suspicious object/person instead of saving the whole frame.

Although researchers have put a lot of effort into this area to develop the efficient tracking algorithms, this area still demands attention due to various difficulties associated with it. The first and the most important constraint is real-time processing power i.e., visual object tracking algorithm should be running in real-time for real-time applications. There is always a tradeoff between the efficiency of the algorithm and time it takes to execute. The second difficulty which most of the algorithms face in real-time applications is severe occlusion. When an object gets occluded with similar object, visual object tracking algorithms easily make a mistake and start tracking the wrong object after the occlusion. So, in this study our main objective is to design an efficient visual tracking algorithm which can consider the aforementioned problems.

1.5 Scope and contributions

This thesis presents the efficient visual tracking algorithm for single object tracking problem i.e., the designed algorithm is capable of predicting the state of the object during the occlusion in collaboration with the estimated position of the object. It is further investigated

that peak correlation score alone is not good enough to detect the heavy occlusion, motion blur, background clutter, out-of-plane rotation, and deformation. Therefore, peak to side lobe ratio is incorporated with peak correlation to detect the occlusion and other issues mentioned above. Furthermore, most of the tracking algorithms deviate from the actual target because of the flawed input. This happens when tracker start tracking the wrong target. In this scenario, the tracker should be capable of detecting the erroneous input and should immediately stop updating the model. We develop a scheme that can detect the erroneous input and stop updating the tracker with erroneous input. Finally, the kernelized correlation filter tracker is modified to increase its efficiency.

In bullet form the main contributions of this study are given below

1) A novel interpretation of difference of the peak correlation (DoP) between the current and previous frame is presented in this study. Negative DoP tells that object is being got corrupt. When the difference between the current and previous frame is positive this tells that the object is coming out of occlusion/deformation.

2) This study presents a novel reliability detection module based on hybridization of average peak correlation energy and confidence of squared response map (CSRM).

3) Novel Adaptive learning rate strategy to prevent the model from being perverted. We update the target model with high learning rate when the APCE is high whereas learning rate is adjusted as per value of APCE which also tell us about the confidence of the tracking result.

4) Prediction is critical step when the object is not visible in a frame. Estimation is also equally crucial during regular tracking. By introducing prediction-estimation collaboration scheme, we achieved better tracking efficiency in distance precision and overlap threshold.

5) The proposed study suggests avoiding template update under erroneous input gives significant improvement in tracking accuracy to handle drift problem.

6) It is shown that peak correlation score alone is not good enough to detect heavy occlusion, motion blur, scale variation, background clutter, out-of-plane rotation and deformation. We computed multiple cues from the response map which includes peak correlation, average peak correlation energy, peak to side lobe ratio, the confidence of squared response map, and novel difference of peak correlation between two consecutive frames. Each cue gives different insight about the target of interest, which helps in accurate occlusion detection and recovery of the target.

7) State-of-the-art algorithm Kernelized Correlation Filter cannot handle the scale variations. This study provides an efficient scale handling strategy for KCF to cater to scale variations.

8) Comprehensive evaluation and analysis of proposed algorithms with state-of-the-art methods on accepted datasets are carried out.

1.6 Research problem statement

Most of the work in the field of VOT is usually based on different assumptions such as single-camera, single-target, model-free, short-term, causal tracking, and limited length of the video etc. Suppose we neglect all or any one of these assumptions tracking becomes a challenging job. The more issues are in a video sequence the more is difficult to track the

object. So, to design a robust tracking algorithm, it must accurately track the object regardless of the changes in appearance model and length of the video etc. In recent literature, several algorithms have been developed to utilize complementary cues for robust visual tracking. However, the robustness of these trackers is still limited in challenging scenarios such as dramatic illumination variation, motion blur, complex object motion and heavy occlusion.

In this study, a visual tracking algorithm to improve the robustness against major tracking issues is presented. The occlusion problem is addressed by a collaboration of prediction and estimation algorithms. In adaptive short term tracking algorithms target model is updated in each frame. When video become long adaptive tracker forgets the actual representation of the model and drift problem starts arising. This problem is addressed by designing and integrating efficient algorithms that keeps the target's long-term memory and adaptivity.

1.7 Research objectives

Many different visual object tracking algorithms have been proposed during the last decade. Most of these algorithms suffer from slow processing. To increase the processing speed, the correlation filter-based tracking algorithm with kernel tricks have been proposed. These algorithms are fine for fast processing but suffer from drifting problem. Our main objective is to increase the robustness and accuracy of the correlation filter-based tracking algorithms.

1.8 Structure of thesis

The rest of the thesis is organized in the following manner.

Chapter 2 at the beginning, describes the general background of visual object tracking, focusing on the correlation filter-based tracking methods.

Chapter 3 describes the main framework for the proposed work and methodology for the main contributions of the thesis.

Chapter 4 starts with the introduction to standard datasets and their attributes in detail. After this comprehensive analysis of results is presented. Both the qualitative and quantitative analyses are presented at the end of this chapter.

Chapter 5 contains the conclusion and discussion about the future work of this study.

Chapter 2.

Literature Review

This chapter initially gives the background of general visual object tracking methods. After discussing the conventional methods, correlation filter-based tracking algorithms are discussed in detail with the background.

To understand the contribution of this thesis, it is necessary to have a concept of how discriminative learning algorithms are usually used in computer vision. We provide a broad overview in Section 2.1, which can be avoided by someone already familiar with this setting. Sections 2.2 and 2.3 will then refer to the sampling problem and concentrate on earlier work that is more closely related to the theoretical framework we develop. Literature reviews of specific applications are within each chapter of this thesis, where they are the most appropriate.

2.1 Image recognition

Image classification is considered the most directly formulated learning problem in image recognition [23], [24]. Let us suppose we have an image containing a single object (may have multiple objects). To classify the object from discrete set of classes like donkeys, vehicles or humans is known as image classification. Event classification and fine-grained categorization are related forms [24]. Rather than learning a model using unrefined pixels, the input to the learned model is usually a representation obtained employing a multi-stage pipeline, intending to exhibit invariance to several confounding factors. The foremost step is typically the extraction of regional features over a grid of places in the frame, such as Histograms of

Oriented Gradients (HOG) [25], Scale Invariant Feature Transform (SIFT) [26], or Local Binary Patterns (LBP) [27], to cite only some common descriptors. They are nearby invariant to brightness and light changes, since most are built on edge detection, and show some invariance to regional deformations, by calculating statistics over small regions. That is why, such features are commonly used as a first processing step in virtually all image recognition problems, not just classification. To explain this point, we should mention that this is the case for most experiments in this thesis, which are based on HOG features. A bit more specific to classification is a coding or pooling stage, which calculates global statistics to form the final representation of the image.

Examples include vector quantization or bag-of-words models [28], spatial pyramids [29],[30], Fisher vectors [31] and Vector of Locally Aggregated Descriptors (VLAD) [32]. The global aggregation subtly yields some invariance to geometric transformations and distortions. A discriminative learning procedure is then trained to predict the image class from this representation. As the output consists of discrete classes, the model in this case is known as a classifier. Learning algorithms normally require a large dataset to learn the model parameters, in this case images and ground-truth. Another possibility is to classify whether the object is present in the image frame or not i.e. classification between object and non-object classes. Then algorithm may be used to find the presence of an object at several different location of an image i.e. performing object detection [33], [11], [34], [35]. Searching the object at many different locations increases the computational cost. That is why, these types of techniques are not common in object detection. Most of the good detectors use one or more simple linear model over HoG features, evaluated in a sliding-window manner and at multiple scales [33],

[11], [34], [35]. Sliding-window detectors were given by the well-known Viola-Jones detector [36]. Large amounts of negative samples are also collected using a sliding window, which is the requirement of any good detector. This means that negative samples are related by translation and can be instantiated as virtual samples. This fact is exploited in [37].

Let us consider a single object tracking problem i.e. tracking an object given only its initial position and size in first the frame [38], [39]. It can also be considered as detection problem using the procedure discussed above. When the object is re-detected in a new video frame, the model needs to update itself for the changes in object display/structure. Hence, we can say that object tracking is simply an online learning problem. while detection and classification is a batch learning problem. The samples obtained in a next frame are also obtained by translation, and since they all belong to the same image, we can make some simplifying assumptions in our analysis of virtual samples. Predicting other extrinsic aspects of an object's appearance is usually called pose estimation [40], [41]. They may include rigid pose parameters, such as an object's rotation or position relative to the camera, either in 2D or 3D [42]. They may also include non-rigid deformation parameters, such as the relative angles of a person's joints [40]. It is possible to learn a model that predicts the pose directly, as real-valued, continuous variables [40], [41], [42]. Another approach is to discretize it into a set of poses, and learn a classifier to identify each pose [34], [43]. This method can more directly benefit from the advances in classifier and detector learning. It also makes it easier to trade off computation (increasing the number of discrete poses) for increased accuracy.

2.2 The sampling problem

When applied to computer vision, a serious challenge for learning is what we will call the "sampling problem". It is mostly an issue of exploiting prior knowledge well. Consider an image that will be used as a sample for learning. Most of the time, any subregion of that image is an equally valid sample. This is especially true for negative samples (i.e. samples that do not contain an object of interest). Thus, a single image can be a virtually limitless source of samples.

Traditional methods deal with this fact by selecting a limited number of samples per image, due to hardware limitations on available memory [33], [11], [44]. The most straightforward method is to simply select the samples randomly, a technique that is most prevalent in tracking applications due to their time-sensitive nature [45], [46]. On the other hand, detector learning mostly relies on hard-negative mining, performed offline [11]. It consists of first training an initial detector using random samples (similarly to tracking). This detector is then evaluated on a pool of images, and any wrong detections (named "hard-negatives") are selected as samples for re-training. Hard-negative mining is a very expensive process, but crucial for good detector performance.

A similar technique is also used in tracking, where detection mistakes are found using a set of structural constraints [45]. A related issue can also occur when evaluating a detector. To localize an object, the learned model is evaluated over many subregions of an image. The amount of computation is proportional to the number of subregions considered, mirroring the sampling problem in learning. Several ideas have been proposed in the literature to address

this problem. One of them is to use branch-and bound to find the maximum of a classifier's response while avoiding unpromising candidate regions [47]. Unfortunately, in the worst case the algorithm may still have to iterate over all regions. Though it does not preclude an exhaustive search, another notable optimization is to use a fast but inaccurate classifier to select promising regions, and only apply the full, slower classifier on those [59, 139]. A related method can quickly discard regions (and thus their subregions) for which the evaluated score will be considered too low [48]. However, it is formulated only for distances between image pairs. Although it may not be apparent at first, virtual samples provide an elegant solution to the sampling problem, making it more amenable to analysis. Subregions of an image extracted at slightly different locations are related by translation. One may approximate them from one subregion by generating virtual samples by translation. The approximation is accurate for most pixels, differing only at the borders. Virtual samples to approximate learning with all possible subregions of several images, which if done naively would be impossible using current hardware is proposed in [49], allow training with all virtual sample translations at a fraction of the computational cost of standard methods, such as hard-negative mining.

2.3 Fourier-domain methods

The recent success of correlation filter tracking motivated us to research this direction [50], [51], [5], [4]. Correlation filters have shown promosing results in term of computational cost i.e., they can process hundreds of frames per second, but using only a fraction of the computational power. This is because convolving two images is simply equivalent to an element-wise multiplication in frequency domain. Thus, by formulating their objective in the

Fourier domain, they can specify the desired output of a linear model for several translations, or image shifts, at once.

Since long been, frequency-domain methods have been used to compute the fast convolution, and were used recently to speed up the detectors at detection time [52]. These Fourrier transforms were also used to speed up the training process of detectors i.e., by modifying an SVM solver with a more efficient sub gradient computation [39]. Fourier domain approach can be very efficient because of the several decades of research in signal processing [50]. At the same time, it can also be very limiting. We would like to concurrently leverage more recent advances in computer vision, such as more effective features, large-margin classifiers or kernel methods [53]. This hinted that a deeper connection between learning algorithms and the Fourier domain was necessary to overcome the limitations of direct Fourier formulations.

2.4 Tracking learning and detection (tracking by detection)

This method considers object tracking as a detection problem in every frame. To make the correlation filter adaptive to the appearance changes of the target of interest, recently proposed methods draw positive and negative sample around the expected target to update the classifier discussed in [3]. However, slightly erroneous labeling of samples accumulates over time and the tracker starts drifting. This problem is known as sampling ambiguity. To handle it many methods have been proposed such as ensemble tracking [54], randomized ensemble tracking [55], adaptive randomized ensemble tracking [56], online multiple instance learning [44], and transfer learning-based tracking [57], [58].

Another problem, with the approach explained in the previous paragraph, is tradeoff between stability and adaptivity. The tracking schemes has been decomposed into three modules, i.e., training, learning and detection given in [45], [59] to keep the system stable and reasonable model adaptivity. The basic idea in the subsequent method is to update the detector with conservative rate using the extra sample obtained from the results of aggressively updated tracker. This online detector can be used in case of occurrence of tracking failure. Examples of such tracker are given in [13], [9], [14]. Online detector for reinitialization of tracker in case of tracking failure is also proposed in [3]. The detection module is activated only if the response is lower than the specified threshold.

Our proposed tracking method also uses a support vector machine-based online trained detector module that differs from the already proposed [3], [14] techniques. We activate the support vector machine-based detector module based on two parameters rather than only the peak correlation value. In our approach, Adaptive Failure Avoidance Tracking Mechanism using Prediction-Estimation Collaboration (AFAM-PEC) response map is utilized to calculate the peak to side lobe ratio along with peak correlation value.

2.5 Correlation filter tracking

Correlation filters are applied in many application areas like object detection and recognition [60]. This operator works as element-to-element multiplication in the frequency domain, and researchers have applied correlation filter extensively to visual object tracking in the last decade due to its less computational cost attribute.

The minimum output sum of squared error (MOSSE) filter is proposed in [4] track monotonic images, where the filter is updated on every frame. This filter is computationally inexpensive having a processing speed of more than a hundred frames per second.

Kernelized correlation filter is proposed in [12], [61], which employs the properties of circulant matrices for extreme fast learning and detection with the help of fast Fourier transform. Efforts have been made to enhance the tracking performance by using correlation filters.

Examples of algorithms based on correlation filter includes multi-channel filters [61], [62], [63], spatio-temporal context learning [16], scale handling and estimation [64], [65], [51] and spatial regularization [66], [67], [68]. Most of these techniques are very good in adopting the fast-changing appearance of the model. Still, due to the non-availability of long-term memory of target appearance, these techniques are susceptible to drift in case of occlusion and out of the view movement of target object. This problem is solved by keeping the long-term memory of the target and deploying two filters, one for short term memory and the other for long-term memory [3]. At the same time this increases the computational cost, and more memory will be consumed.

Compressive tracking algorithms are also presented in recent years which extracts the features from multi-scale feature space whose basis function do not depend upon the data. One of the examples of these types of trackers is given in [17].

Unlike existing techniques that employ only correlation filter for translation estimation even during occlusion, we introduced the predictor module, to handle the drifting/tracking

failure in case of occlusion, motion blur and out of the view movement of the target. In our approach (AFAM-PEC) predictor is incorporated with the short-term memory correlation filter.

Peak to side lobe ratio and peak correlation score from response map are calculated to predict an objects's occlusion/motion blur/out of view movement. Based on these two parameters confidence score is calculated, which will decide the reliability of the tracking result for that specific frame. Short term memory filter will stop updating its weights if tracking reliability is less than a certain threshold say tr. Measurement follower predictor is used to predict the next position of the object during occlusion time. Once the tracking result reliability approaches specified threshold say t_s, again short-term memory filter is activated to estimate the next state of the object.

2.5.1 Minimum Output Sum of Squared Error Filter

Originally correlation filters used simple templates and mostly failed when applied to tracking applications. Later, the minimum output sum of squared error filter was presented for visual tracking applications [4]. This filter showed prominent results in term of computational time and tracking efficiency. This filter minimizes the mean squared error of actual output and required output. Mathematical MOSSE filter can be given as follows:

$$w = \underset{w}{\operatorname{argmin}} \frac{1}{N}\sum_{1}^{N} \|w \otimes x_t - g_t\|_2^2 + \lambda\|w\|_2^2 \qquad (2.1)$$

where w is the correlation filter that minimizes the sum of squared error between the actual output and the desired output, x_t is the $d \times 1$ vector version of the training example, and g_t is

the $d \times 1$ desired correlation output for that training example. Typically, g_i is chosen to be a Gaussian function centered at the origin with a small σ for positive training examples, and all zeroes for negative training examples. The parameter λ is for regularization. Note that the original formulation in [1] does not include the regularization term; this is equivalent to $\lambda = 0$, but we include it for generalization. We can express Eq. 2.1 in the frequency domain, and we can find the minimum by solving $grad(\hat{w}) = 0$.The canonical form is written as

$$w = \frac{\sum_{i=1}^{N} \hat{f}_i^* \odot \hat{g}_i}{\sum_{i=1}^{N} \hat{f}_i^* \odot \hat{f}_i + \lambda} \tag{2.2}$$

where dot the operator represents the Hadamard product. Also note that the parameter λ has subsumed a factor of N. When used in scenarios that require incremental learning, such as visual tracking tasks, the MOSSE filter can be updated as a linear combination of the previously learned filter and a filter built on the new training examples. Given a filter $\hat{w}_N$ learned on the first N training examples, we can simply add an element to the summations required in Eq. 2.2. Assume that

$$w = \frac{\sum_{i=1}^{N} \hat{f}_i^* \odot \hat{g}_i}{\sum_{i=1}^{N} \hat{f}_i^* \odot \hat{f}_i + \lambda} = \frac{\hat{a}_N}{\hat{b}_N + \lambda} \tag{2.3}$$

When a new input $\hat{f}_{N+1}$ becomes available, we can update $\hat{a}_N$ and $\hat{b}_N$ with the following updates

$$\hat{a}_{N+1} = (1 - \eta)\hat{a}_N + \eta(\hat{f}_{N+1}^* \odot \hat{g}_{N+1}) \tag{2.4a}$$

$$\hat{b}_{N+1} = (1 - \eta)\hat{b}_N + \eta(\hat{f}_{N+1}^* \odot \hat{f}_{N+1}) \tag{2.4b}$$

where $0 \leq \eta \leq 1$ is a parameter controlling the learning rate.

Smaller values of η correspond to slow adaptation, whereas larger values of η correspond to more aggressive adaptation. The MOSSE filter is used in visual tracking because the update scheme in Eq. 2.4 allows a tracking system to update the target model quickly. Bolme et al. provide a qualitative comparison of the MOSSE tracker's accuracy compared to other trackers at that 24time, while boasting an impressive 669 FPS. At such fast speeds, the MOSSE filter became a viable starting point for subsequent trackers that could be developed to be more robust and accurate at the expense of speed while remaining faster than real-time (30 FPS).

2.5.2 Kernelized Correlation Filter

The tracking algorithm [7] builds on the MOSSE filter concept [4] by extending the filter to non-linear correlation. Linear correlation between a CF template and a test image is the inner product of the template w with a test sample z for every possible shift of the test sample z. Instead of computing the linear kernel function $w^T z$ at every shift of z, KCF computes some non-linear kernel κ(w, z) = $\varphi^T(w)\varphi(z)$ where κ represents a kernel function that is equivalent to mapping w and z into a non-linear space with the lifting function ϕ(·).

In one sense, KCF can be viewed as a change away from linear correlation filters. Still, it can also be seen as an efficient way of solving and testing with kernel ridge regression when the training and testing data is structured in a particular way (i.e., a circulant matrix). Henriques et al. derive KCF from the standard solution of kernelized ridge regression. For learning w, we assume the training data $X = [x_0. x_1, \ldots., x_{d-1}]$ is a d × d matrix where x_k contains the same elements as x_0 shifted by k elements. The solution to kernelized ridge regression is given by [3].

$$\alpha = (K + \lambda I)^{-1} g \tag{2.5}$$

where K is the kernel matrix such that $K_{ij} = k(x_i, x_j)$, I is the identity matrix, λ is the regularization parameter, g is the desired correlation output, and α are the dual-space coefficients.

The dual-space coefficients allow us to rewrite the original template w as

$$w = \sum_{i=1}^{N} \alpha_i \varphi(x_i) \tag{2.6}$$

Where the kernel function κ(w, x) treats all data elements equally, and kernel K and the coefficients α can be computed efficiently in the Fourier domain as follows:

$$\hat{\alpha}^* = \frac{\hat{g}}{\hat{k}^{xx'} + \lambda} \tag{2.7}$$

Where $\hat{k}^{xx'}$ represents the first row of the kernel matrix K which contains the kernel function computation of x_0 with all possible shifts of another data sample denoted x' ; either x_0 in the training phase, or some test sample z in the testing phase. This idea is getting closer to the use of the Fourier domain to compute linear correlation efficiently. With non-linear kernels, Henriques et al. show that all elements of $\hat{k}^{xx'}$ can be computed efficiently. As an example, the Gaussian kernel $\kappa(x, x') = \exp\left(-\frac{1}{\sigma^2}(\|x\|^2 + \|x'\|^2\right)$ can be computed as

$$\hat{k}^{xx'} = \exp\left(-\frac{1}{\sigma^2}(\|x\|^2 + \|x'\|^2 - 2\mathcal{F}^{-}(\hat{x} \odot \hat{x}'^*))\right) \tag{2.8}$$

Where $\mathcal{F}^{-}$ represents the IDFT.

Just as in the case of the linear kernel, computing the Gaussian kernel in the Fourier domain reduces the computational complexity, although there are additional DFT/IDFT operations called compared to the linear kernel. During training, this kernel is computed for learning the coefficients $\hat{\alpha}^*$ as shown above in Eq. 2.7, and when testing, the correlation is computed as

$$\hat{g}' = \hat{\alpha} \odot \hat{k}^{xz} \tag{2.9}$$

where the IDFT of $\hat{g}'$ will produce the non-linear correlation in the space domain.

The KCF tracker utilizes these non-linear kernels to achieve performance improvements over the MOSSE filter. It operates similary, with an update scheme in the same spirit as Eq. 2.4. One important distinction between the MOSSE tracker and the KCF tracker is that the MOSSE tracker derives and stores a correlation filter. In contrast, the KCF tracker computes and stores the dual space coefficients and, necessarily, the training examples. As tracker continues through a video sequence, the tracker could retain multiple training images, but this would result in progressively slower tracking as the computational demands in solving for the kernel matrix $\hat{K}$, and subsequently, $\hat{\alpha}$ as shown in Eq. 2.7, grow with the number of images stored. Rather than attempt to store multiple distinct training images or to discard data from old frames entirely, the KCF tracker stores a single training image $\tilde{x}$ that is a linear combination of previous images, so that

$$\hat{x}_k = (1-\eta)\hat{x}_{k-1} + \eta x_k \tag{2.10}$$

where x_k is the image patch in the k^{th} frame.

When the KCF tracker was first introduced, it exhibited better accuracy than other trackers at that time, while reporting speeds greater than 150 FPS [61]. This combination of accuracy and speed made it a popular tracker to improve upon in several ways. Finally, it is important to note that the KCF is largely a reformulation of the CSK tracker introduced earlier by Henriques et al. [12]. One of the biggest changes between the CSK and KCF trackers is the addition of multi-channel features, which were previously introduced for CFs [69], [58]. Henriques et al. incorporate multi-channel features as a straightforward way to improve performance, where each channel is treated independently, and the correlation planes of each channel at test time are summed.

2.5.3 A Simple Correlation Filter Tracker

The previous two sections discussed the design principles of the two CFs that appear in almost all CFTs. This section outlines how either the MOSSE filter or KCF can be implemented within a simple tracker. The simple tracker explained in this section can be considered a baseline tracker that other CFTs modify, but many details are first seen in the KCF tracker. Modifications often include swapping out particular components of this generic tracker, but other trackers may modify more significant portions of the tracking workflow; these changes are discussed at length in Sec. 2.6. As we know that the input to a tracker, along with $I = \{I_0, I_1, \ldots, I_n\}$, is only the first frame of a video with a rectangular bounding box $b_0 = [x_0, y_0, w_0, h_0]$ denoting the target region, and the output of the tracker is rectangular bounding boxes $b'_k = [x'_k, y'_k, w'_k, h'_k]$denoting the target location estimates in the rest of 27the frames of the video. In the first frame, we extract features from the given bounding box. It is important to detail how this training is done. Typically in CF applications, the template and the test image

are zero-padded when performing the FFT [70]. The zero-padding is assumed to remove the circular convolution effects introduced by the DFT [71]. However, most CFTs do not perform zero-padding before computing the FFT of any space domain templates or image patches. Instead, the CF is computed from a region larger than the actual target; in some trackers this padding results in a CF with a width and height 2× larger than the original target scale [w_0, h_0][3].This extra padding is done in combination with applying windowing – usually a cosine window – to both reduce the impact of the circular convolution and to emphasize the features that are within the original target region (and within the target region, the windowing emphasizes the center of the target even more).

This design decision has tradeoffs: the larger CF allows the tracker to learn against some background information implicitly, and the windowing does reduce aliasing effects, but they are not removed entirely (and zero-padding this larger CF would likely reduce the speed significantly). It's important to note that incorporating the background into the CF training makes more sense in the tracking application; unlike tasks like ATR or other single-image object detection tasks, we know we will have to distinguish the target from a similar background in subsequent frames; this is not the case in other detection tasks.

From the first frame, we have an initial CF to detect the target throughout the video. In subsequent frames, the tracking process can remain relatively simple. An image patch centered on the previous estimated target location is extracted from the new frame. This patch is usually the same size as the padded CF. The same feature extraction and windowing is performed. We take the DFT of the feature representation of the target, and perform the correlation between the image patch and the CF. We take the IDFT of correlation output, and the maximum

correlation value determines the center position of the target in the correlation output plane. If the target's scale is estimated, it may be done at this stage, or it may be done jointly with the translation estimation of the target. More details regarding scale estimation can be seen in Sec. 2.7 (the Original KCF tracker does not perform any scale estimation). For trackers that perform redetection and/or failure detection, this is usually when that tracking element is exercised (the KCF tracker does not perform any redetection).

Following the estimated target position (and scale), the CF must be adapted to the new target information. At the new target location (and appropriate scale), a final image patch is extracted4 and features are again extracted and windowed. This image information is incorporated into the filter design. The MOSSE and KCF filters are designed so that this can be done with a simple linear combination of the previous CF and a CF designed solely on the new detection, i.e., $\tau_t = (1-\lambda)\tau_{t-1} + \lambda\tau_{new}$, where τ denotes whatever filter design is used, and λ denotes the adaptation rate that balances the previously learned model τ_t and the information from the new detection, denoted τ_{new}. The details of the update as well as the value of λ will vary from tracker to tracker as CF designs vary, but nearly all newer CFTs will update their filter model. The two-step process of detect-update will continue through the duration of the video. An overview of this entire system framework is shown in Fig. 2.1. We note that certain design decisions that are precluded in the Online Tracking Benchmark (OTB) and Visual Object Tracking (VOT) benchmark challenges. Both benchmarks prohibit revising old detection outputs based on more recent frames. Additionally, specific pretrained models are not allowed on a per-video basis. However, the entire dataset has no prohibition on tailoring

certain tracker parameters (e.g., amount of padding, adaptation rate λ, or CF specific parameters).

2.6 Improvements for Correlation Filter Trackers

2.6.1 Feature Representations for Correlation Filter Trackers

Traditionally, CFs assumed scalar or single-channel features, e.g., grayscale intensities when operating on images. This was the case for much of the period before CFTs being introduced. CFs that accommodated other features, called vector CFs [69] or multi-channel CFs [61], have been developed in more recent years. These CF designs were like the MOSSE formulation, in terms of minimizing the MSE of the correlation output plane. The work presented by both [69] and [58] use the cross-spectral energy between different feature channels, and both choose histogram of oriented gradients (HOG) features to illustrate the new CF designs. As was discussed in Sec. 2.3.2, the KCF tracker uses multi-channel HOG features, but treats them independently. While treating each feature channel independently reduces the computation time, it ignores the possible interactions between feature channels and effectively assumes that all feature channels are independent. We note that the choice to treat each feature channel independently is the 30prevailing choice in CFT designs.

Histograms of Oriented Gradients

From the original MOSSE and CSK trackers that used only scalar features, several feature descriptors have been explored in CFTs. As discussed above, the KCF tracker was the first to introduce HOG features to CFTs. HOG features were originally introduced for pedestrian detection but have become a popular feature descriptor in a range of object detection tasks

[18]. HOG descriptors aim to capture edge features of a given target. Since the first use of HOG in CFTs, a number of subsequent CFTs use HOG, either as the only feature descriptor or in conjunction with other feature descriptors [7],[72][73][15][72][25]. The KCF tracker used a HOG cell size of 4 × 4 and retained all 31 feature channels, meaning that a HOG descriptor for an image patch of size w × h would be $\frac{w}{4} \times \frac{h}{4} \times 31$. Most trackers use the 4 × 4 cell size from the KCF tracker, but some trackers change this; some CFTs use a 1 × 1 cell size [65], a 6 × 6 cell size [20], and another that uses 2 × 2 cell size for small targets and the 4 × 4 cell size for larger targets . Most trackers retain all 31 feature channels. The decision regarding cell size becomes a familiar tradeoff; smaller cell sizes produce denser feature descriptors but reduce the speed of the tracker; using 31 HOG feature channels requires 31 FFTs. Larger cell sizes will keep the tracking speed much faster, but may not characterize the target well, particularly smaller targets. Overall though, HOG features can be computed quickly (independent of subsequent FFTs), perform much better than grayscale intensity features [74], and do not appear to slow tracking down at all when done at a cell size of 4 × 4 [74]. One final note regarding HOG features is that using a cell size of c × c means that the smallest detected target translations will be c pixels by default. The original KCF tracker does not address this, and therefore all estimated target translations are multiples of 4 pixels. A modified version of the KCF tracker uses sub-cell peak estimation to estimate smaller target translations than the HOG cell 31size [75].

Color features

While HOG descriptors capture the edge characteristics of a target, other features attempt to capture the color information of the target object. The target object has a distinctive color in

many videos, e.g., a track and field athlete wearing a distinct jersey. If the target is the only yellow object in the video, it certainly seems straightforward to simply find the yellow blob in each frame. The Adaptive Color Tracker (ACT) was the first CFT to use color information for feature descriptors of the filter and target image patches [10]. The authors explore the effectiveness of several color spaces, e.g., RGB, LAB, HSV, and others. Their investigation shows that color name attributes perform best. Color names are a higher dimensional representation of colors based on human perception. Unlike other color spaces which have a mathematical formula to convert from RGB color space, small ranges of RGB values are mapped to a probabilistic 11-dimensional vector that sums to 1, where each value corresponds to human perception of black, blue, brown, grey, green, orange, pink, purple, red, yellow, and white. Color names, like HOG descriptors, had previously been used in other computer vision tasks [76]. Since the publication of the ACT tracker, a number of CFTs have used color names jointly with HOG descriptors [74], [77]. The ACT tracker also proposes dimensionality reduction for the color names; this results in a 25% increase in the frames per second (FPS) while only slightly reducing accuracy.

While the color name features are a pixel-wise descriptor, other trackers use color information in a different manner. The Sum of Template And Pixel-wise tracker [73] uses the first frame to learn which RGB values are representative of the target. In subsequent frames, per-pixel scores based on RGB values are smoothed out over a region equal to the target size to produce a color response plane. The amount of smoothing precludes a sharp peak from appearing within the color response, but it is used as a 32 complement to a CF built with HOG features. The HOG result will often produce a much sharper correlation peak, while the color

information serves to reinforce or alter less sharp peaks that would correspond to less confident detections from the HOG features. The two output responses have different shape characteristics derived from different information (color vs. texture), and the overall result is stronger.

Much like HOG descriptors, color features are very quick to compute. Most color spaces have 3 channels, while the color names have 11 channels, which does result in slower performance (while HOG features often make up for 31 feature channels by reducing the spatial resolution, color features are typically of the same spatial resolution as the original target).

Deep-Learned Features

In recent years, deep convolutional neural networks (CNNs) have come to supplant "handcrafted" features like HOG in computer vision tasks [78], [79], [80], [81], [82]. While hand-crafted features like HOG are computed based on what researchers expect to be salient feature outputs for discrimination, deep neural networks (NNs) are expected to learn discriminative features on their own, given sufficient training data. Just as deep features followed the introduction of a range of hand-crafted features in domains such as object classification and localization, CNNs are being introduced to visual tracking shortly after hand-crafted features. The visual tracking problem is characterized by the lack of target data prior to the beginning of tracking. This immediately rules out training a CNN from scratch; instead, most CFTs that employ CNNs depend on a pretrained model, typically either AlexNet [78] or VGGNet [83]. At a high level, CNNs take an input image and, over successive layers of convolutions with filter banks and spatial pooling, learn feature representations that capture

elements ranging from low-level textures to image classifications. For the simplest use in visual tracking, the CNN can be considered a static feature extractor, like HOG or color features. This most basic approach is shown in work by Danelljan et al. [84], simply called the DeepCFT. The authors build a CF using the output from each convolutional network in VGGNet, which takes an input image patch of 224 × 224 × 3 (all targets are resized to fit the pretrained network) and outputs a 109 × 109 × 96 descriptor from the 1st convolutional layer, a 26 × 26 × 256 descriptor from the 2nd convolutional layer, and 13 × 13 × 512 descriptors from the 3rd, 4th, and 5th convolutional layers. The authors show that the best performance is obtained when building the CF using the 1st output layer, and in fact the performance drops off each successive layer until the 5th layer, which performs only 3rd best. The assumption is that the deeper layers do not provide enough spatial resolution; there is roughly a 17× reduction in the spatial resolution from the original patch to the 3rd layer. Most importantly, the authors show that the CFs built from the CNN's output outperforms comparable CFs built jointly from HOG and color name features. From this simple approach to building a deep CFT, more advancements have been made. Rather than just using the output from one layer, other works have combined the outputs from different layers [85], [86], [87].

The Multi-Level Deep Feature Tracker (MLDF) goes beyond just using a pretrained network and actually uses the current track information to train the CNN to adapt to the target appearance and its surroundings, rather than just keeping the starting VGGNet. A look at recent benchmark performance shows that deep features are used in many of the most accurate trackers [88]. However, the use of deep features does come at the cost of speed; CFTs using CNNs are typically much slower than trackers using hand-crafted features. Despite this, visual

tracking may follow a similar trend as other computer vision tasks that have become more and more dominated by deep networks.

2.7 Target scale estimation

In their simplest construction, CFTs simply estimate the translation of a target; the 2D correlation output plane provides no insight into the changing scale of a target. Accurately estimating target scale has multiple benefits; it directly affects how tracker performance is quantified in benchmark evaluations, and beyond benchmarks it can provide important information in real-world applications. Along with being valuable in and of itself, accurately estimating scale allows a tracker to adjust its own tracking procedure to adapt to the changing target, thus reducing the possibility of drifting off of or losing the target entirely. With the possible intrinsic and extrinsic benefits of accurate scale estimation, there has been a good deal of work in adapting CFTs to scale variation.

2.7.1 Exhaustive scale search

Perhaps the earliest CFT to address scale estimation was the Discriminative Scale Space Tracker (DSST)[89], [90] . DSST shares many similarities with the KCF tracker but adds a scale estimation component following the translation estimation of the target. Following the translation estimation to determine $[x'_k, y'_k]$, the tracker extracts image patches at S scales. For each scale $\eta \epsilon \left[\left\lfloor -\frac{s-1}{2} \right\rfloor, \ldots, \left\lfloor \frac{s-1}{2} \right\rfloor\right]$, DSST extracts an image patch of size $a^n w'_{k-1} \times a^n h'_{k-1}$, where a is the scaling factor between adjacent scales and $[w'_{k-1}, h'_{k-1}]$ is the previously estimated target size. Similar to the process for estimating the target translation, a separate CF designed for estimating scale is correlated with the feature descriptors extracted at each of the

S scales (DSST chooses S = 33). Because the target centering is roughly accomplished, the correlation output is limited to a S × 1 output. Just as the CF for translation estimation is built in a way to favor smaller translations, a 1D Gaussian windowing is applied to the S × 1 scale correlation output to favor smaller scale changes. This, along with a conservative scale factor of a = 1.02 results in small estimated scale changes from frame to frame, which is consistent with target behavior in most applications when the video's frame rate is 24-30 FPS (as is the case of nearly all benchmark videos). When the target scale is estimated, the scale filter is updated. In subsequent frames, the input image is resized according to the current scale for the estimation of the target translation. Finally, we note that while the translation filter in DSST uses 1 × 1 cell size HOG features, the scale filter uses PCA-HOG features [11] with 4 × 4 cells. The justification for using a larger cell size is that pixel-wise estimation is only a concern for the target translation.

DSST estimates the target bounding box in two steps, first by estimating the translation [Δx, Δy] via a 2D correlation, and then jointly estimating [Δw, Δh] by estimating the change in scale via a 1D correlation. The authors of DSST also explore estimating the translation and scale together by learning a 3D scale-space CF. They find that this 3D CF is much slower, as would be expected, and also actually performs slightly worse than the sequential translation and scale estimation. Following the findings during the development of DSST, a number of trackers have followed the approach of sequential translation and scale estimation [91], [84]. Still, another CFT does jointly estimate translation and scale [68].

2.7.2 Efficient scale search

DSST and trackers that adopted the same approach look exhaustively over various scales, which may not be the most efficient approach possible. The Multi-Kernelized Correlation Filter (MKCF) trackers [92], [93] seeks to estimate the scale by finding the scale that produces a correlation output peak with the highest PSR in a more efficient manner by performing a line search within a range of scales ±10% of the current scale. This approach assumes that the PSRs across different scales within the search range will 36only have a single local maximum; with multiple local maxima, a suboptimal scale could be chosen. The authors of [61] do not report how often this assumption is accurate. In addition to a line search, MKCF chooses to rescale the features rather than extract features multiple times from different scales. The authors do compare this approach to the "traditional" approach of extracting features from image patches of different scale and show a small gain in the accuracy when doing the faster rescaling of features, though this effect is relatively small.

The trackers mentioned above estimate the scale either exhaustively or with a more efficient line search. The Multi-view Correlation Filter Tracker (MvCFT) [94] reduces the scale estimation to a discrete decision to decide if the target is getting smaller, remaining the same size, or getting larger. Once the target translation is estimated, image patches at the current scale and ±5% are tested against the same CF used in translation estimation. The maximum value from the correlation output plane for each of the 3 scales is taken, and, after the unchanged scale is given a small amount of extra weight, the maximum value of all 3 planes is used to determine if the target is getting smaller, larger, or remaining the same size. If the scale is changing, it is changed by 5%. It can be assumed that very small-scale changes will

not be detected, but CFTs without any scale estimation are robust to minor changes, so effectively ignoring the smallest changes should not be a large concern. The authors claim experimental results support this.

2.8 Parts-based correlation filter trackers

Parts-based models often seek to perform vision tasks by decomposing a large object into smaller pieces that can be operated on independently and then joining the results of the subproblems to provide a coherent result for the entire object. Parts-based models have been used previously in object alignment [95] and object detection [11]. More closely related to CFTs, recent work on object alignment used CFs to detect individual car parts before fusing the result with a deformation model [69].

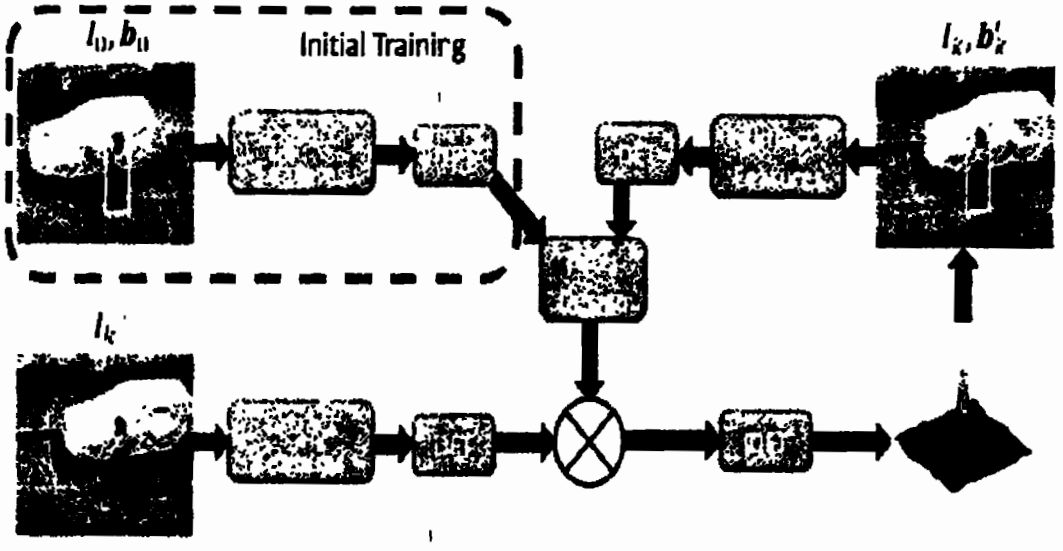

Figure 2.1 Standard correlation filter-based tracking. I_0,b_0 represents the coordinates of the target in the initial frame, I_k represents the next frame, b_k' represents the estimated position of the target in the next frame.

Figure 2.2 Part-based Correlation filter tracking. The top two images show that tracker uses five parts[96] whereas two parts are used in the bottom two images[72] .

Parts-based approaches are designed to handle object deformations, e.g., a pedestrian's changing stride. In contrast, a single rectangular detection window may struggle to characterize all possible poses of the target. Additionally, parts-models are inherently tolerant to partial occlusion; if some object parts are occluded, a detector can still work well based on the strength of the visible parts.

Visual trackers stand to benefit from some of the intrinsic characteristics of parts models, but their application and benefits are not the same as those in single-image object detection. Parts-based object detectors can benefit from target knowledge and possible deformations, e.g., a pedestrian detector can be designed to have a part for each limb. With no prior knowledge of

the target, parts cannot be defined so precisely for trackers; instead, the configuration of generic part is necessary. Still, visual trackers will benefit from robustness to occlusion. Additionally, parts-models can easily be tailored to estimate scale; if parts are drifting farther apart, that alone can be enough to indicate that the scale is increasing.

A tracker proposed in [96] uses KCF filters to localize 5 object parts, each approximately 20% the total target height and width, configured in a cross pattern as shown in Fig. 2.2. The individual part detections are given weights according to two factors: a higher PSR for a part's detector will result in higher weight, and a smaller shift from the previous location will result in a higher weight. The use of PSR is mostly self-explanatory but emphasizing smaller shifts does require justification. The authors argue that detectors for parts that become occluded can possibly detect another unoccluded part of the object; this is part of the risk of defining a generic parts-model for all possible targets. More generally, the justification is that if all part detectors shift equally between frames, it is likely the entire object did, and there will be no net effect of this shifting penalty on the relative weights between parts. If 1 of the 5 part detectors shifts much more than the other parts, it is more likely an error and should be given less weight (although, under this design, an outlier part could possibly shift less; this is not addressed). Once the individual correlation planes are weighted, they are combined to provide a full confidence map. The final target translation and scale estimation are determined by Bayesian inference similar to a previous tracker [97], [98]. The tracker proposed by Akin et al. [72] subdivides targets into only 2 parts: either top and bottom parts for tall and narrow targets or left and right parts for short and wide targets as given in Fig. 2.2. The two parts use KCFs filters to localize their half of the target.The reliability of the 2 part detectors is indicated by

the correlation peak value for each part. Based on these weights, an additional full-target KCF is built as well and performs a full-target target detection centered at the location suggested by the two-part detectors.

The target's scale is estimated by measuring the changing distance between the two parts. When updating the CF models, the correlation peaks are tested against a threshold; if a part detection falls below some threshold, the part model will not update, and if all parts fall below the threshold, the full-target detector will not update. This is meant to avoid updating the models when the target or target part is occluded. Additionally, scale estimation is only performed when all part detections are considered reliable.

2.9 Other visual trackers

In Sec. 2.6, Sec 2.7, and Sec. 2.8, wide range of CFTs and the various improvements they make to the MOSSE and KCF trackers that first used CFs for visual tracking were discussed. However, both prior to introducing the MOSSE tracker and during the continued growth of CFTs in recent years, many other visual trackers that do not use CFs have been developed.

One of the most well-known trackers is the Tracking-Learning-Detection (TLD) tracker introduced in 2012 [45]. As its name suggests, the tracker has three components. A Median-Flow tracker [99] locates the target from frame to frame based on the current trajectory. The detector treats new frames independently of previous frames and can correct failed track. The learner observes both the tracker and detector, and estimates when the detector is making errors. Based on when the learner believes the detector is making errors, it can generate more training data for the detector to improve its performance. The TLD tracker was the third most

accurate tracker when the OTB50 benchmark was published in 2013 with 29 trackers included [100]. The TLD tracker is not evaluated on the most recent VOT benchmarks, but a more recent proposed tracker that fuses the principles of both the TLD and KCF trackers has been proposed [101].

The best performing tracker in the OTB50 benchmark's collection of trackers was the Sparsity-based Collaborative Model (SCM) tracker [15]. Zhong et al. introduce a discriminative and a generative model that learns sparse grayscale features and sparsity-based histograms within a particle filter framework. The SCM combined the two approaches while stating that the generative model plays a more significant role in tracking. The second best tracker reported in the original OTB50 benchmark was Struck [46]. Struck trains a structured output kernel SVM that continuously adds previous detections and hard negatives from the region around each detection, while pruning the number of possible support vectors over time to avoid progressively slower processing times.

Despite the success of the above trackers, many new trackers have been introduced since the OTB and VOT benchmarks essentially regulated the ways trackers operate and are evaluated; nearly all of the most effective trackers on these benchmarks have been developed since the introduction of these benchmarks (and perhaps developed explicitly for the challenges present in the datasets). This is true for CFTs and other trackers.

The best performing tracker on the VOT2015 dataset was the Multi-Domain Network (MDNet) tracker [102]. MDNet pretrained a CNN on an outside set of videos, then combines this pretrained network with a binary classification layer for a test video. Candidate regions are

sampled with varying translations and scale changes relative to the previous target detection. The significance of MDNet is that its success, coupled with being one of the first trackers to use CNNs, likely inspired a growing number of newer trackers that use deep networks. Other CNN trackers include an extended version of the MDNet tracker, with occlusion inference and a scale regression model to refine the output bounding boxes, submitted to the VOT2016 benchmark [88], and the Tree-Structured Convolutional Neural Network (TCNN) tracker [103] that uses a CNN along with a tree structure to capture the multi-modal appearance of certain targets. Another recent tracker exchanges the CNN for a Siamese NN [104].

While deep networks are growing in popularity, other trackers which also performed very well in recent benchmarks. The Edge Box Tracker (EBT) [105] uses an objectness measure to find region proposals within an entire frame, which any object detector can then process. The tracker focuses on finding hard false-positives and re-ranking proposal regions, which can be processed separately. The salient region-based tracker (SRBT) uses color information to segment a target more precisely than a rectangular bounding box; this more precise segmentation determines which regions of the rectangular bounding box contribute to the model update [88]. The geometric hypergraph tracker (GGT) [106] uses a graph structure to capture the relationships between different target parts as correspondences between frames are found and used to find a subset of reliable parts. An extended version of GGT appears in VOT2016 that incorporates the Scale Adaptive with Multiple Features (SAMF) CFT.

Along with original tracking systems, the outputs from multiple trackers can be combined to produce one composite output. The median absolute deviations (MAD) fusion strategy [107] is able to detect outliers, or trackers which have likely failed. Each individual tracker deviates

from the median determines the weight given to that tracker for the final estimate, and outliers are ignored and reinitialized on the new estimated target location. The VOT2016 benchmark contest uses a swarm of KCF trackers and a DSST scale estimation scheme, and outperforms both KCF and DSST [89].

2.10 Summary

This chapter initially presented a general background on how discriminative learning algorithms are used in computer vision applications. After this, some state-of-the-art tracking algorithms with a major focus on correlation filter-based tracking are described. In the end some improvements to correlation filter-based tracking schemes are presented.

Chapter 3 is based on the methodology of this research study. The main components of the proposed work are discussed in detail.

Chapter 4 describes the results and its detailed discussion. Whereas Chapter 5 provides the conclusions of each contribution and future recommendations.

Chapter 3.

Methodology

Our objective is to develop a robust online training-based visual object tracking algorithm to handle the long-term occlusion more effectively than other already proposed long-term tracking methods. Without considering the orientation of the object, tracking is simply the estimation of the translation and scale object [3], [14]. In our proposed framework, the translation estimation is based on the correlation of temporal context and scale estimation is based on the discriminative correlation filter.

In this section, the main components of the proposed tracking procedure are described. First of all long-term correlation tracking [3], [14] is described in Section IIIA. Next, we describe the estimator module [3] in Section IIIB, we use support vector machine classifier. Section IIIC describes the Kalman filter-based predictor. Finally, in Section IIID, predictor strategy is described, which assists the translation estimation filter during long-term occlusion. Different notations and variables used in the following sections are given in Table 4.1.

3.1 Long-term correlation tracking

Correlation filter based tracker calculate the weights w by training on an image patch i of $P \times Q$ pixels to model the target appearance [3], [14], [57], [58], [61], [64], [51], where all the circulant shifts $i_{p \times q}, (p, q) \in \{0,1, \ldots, P-1\} \times \{0,1, \ldots, Q-1\}$, are considered as samples for training with the gaussian function label $y(p, q)$, i.e.

$$w = \underset{w}{\operatorname{argmin}} \sum_{p,q} \left|\sigma(x_{p,q}) \cdot w - y(p,q)\right|^2 + \lambda(w)^2 \tag{3.1}$$

Where σ is mapping to kernel space and λ denotes regularization parameter, which is always greater then or equal to zero. As labeling is not binary , hence w contains the coefficients of gaussian ridge regression model [108]. By using fast Fourier transform the above objective function is minimized to the Eq. 3.2.

$$w = \sum_{p,q} c(p,q)\, \sigma(i_{p,q}), \tag{3.2}$$

where c is calculated by (3.3) using discrete Fourier transform as follows:

$$C = f(c) = \frac{f(y)}{f(\sigma(i).\sigma(i)) + \lambda'} \tag{3.3}$$

f denotes the discrete Fourier transform (DFT) and $y = \{y(p,q) | (p,q) \in \{0,1,\ldots,P-1\} \times \{0,1,\ldots,Q-1\}\}$.

In the new frame, response map over image patch u of size $P \times Q$ is calculated by using inverse discrete Fourier transform as per Eq. 3.4.

$$\hat{y} = f^{-1}\left(C \odot f\left(\sigma(u) \cdot \sigma(\hat{x})\right)\right), \tag{3.4}$$

Where $\odot$ is element-wise multiplication, $\hat{x}$ is learned target appearance model and maximum value of $\hat{y}$ is the new target location. Two correlation filters are trained using single frame, one to model the target appearance solely and other to model the surrounding along with the target. As surrounding information does not change quickly and remains temporally

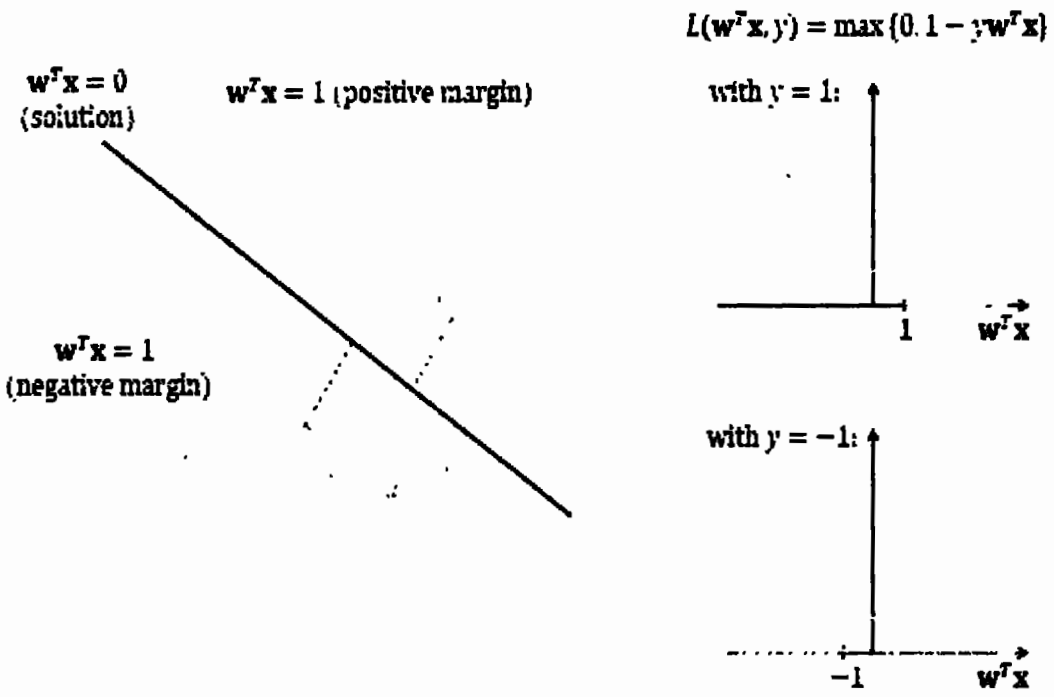

Figure 3.1 Support Vector Machine demonstration. On the left side green circles represent the positive sample whereas, blue crosses representing negative sample. This is the case of the linear separable dataset. The solid black line represents the hyperplane separating the closes positive and negative samples.

stable, it is very useful to differentiate the target from the background in case of occlusion [3], [14], [16]. A weighted cosine window is applied to feature channels to remove the boundary discontinuities of the response map.

Context Regression model $R_{con}$is adaptive to cater the occlusion, abrupt motion and deformation with the learning rate as Eq. 3.5 and Eq. 3.6:

$$\hat{\imath}^t = (1-\beta)\hat{\imath}^{t-1} + \beta i^t, \tag{3.5}$$

$$\hat{A}^t = (1-\beta)\hat{A}^{t-1} + \beta A^t, \tag{3.6}$$

Target appearance regression model R_{tar} is learned from the most reliable and confidently tracked frames. Reliability is determined using maximal value of $\hat{y}$ [3]. Unlike the existing techniques [3], [14] to maintain the model stability in true letter and spirit, two thresholds are defined to update the target regression model R_{tar} using (4). First threshold T_a is on peak correlation value. Second threshold T_{psr}, is on peak to side lobe ration of the response map. If both the criterion are met, then only target appearance regression model is updated using Eq. 3.4 i.e. $max(\hat{y}) > T_a$ & $PSR(\hat{y}) > T_{psr}$. Not that only the peak correlation value is enough to ensure the model stability in case of lorg-term occlusion, as shown in Fig. 11. We update the target appearance regression model only if the tracker results are above the certain reliability threshold i.e. T_{tar} , we keep the learning rate β aggressive.

For optimal scale selection of tracked target, image pyramid technique is implemented using the concept of [34]. If $P \times Q$ is the size of target and N is the number of scales, then

$$\hat{s} = \underset{s}{\operatorname{argmax}}(\max(\hat{y}_1), (\max(\hat{y}_2), \ldots, (\max(\hat{y}_s)), \tag{3.7}$$

Where each $s \in S$, $S = \{a^n | n \left[-\frac{N-1}{2}\right], \left[-\frac{N-3}{2}\right], \ldots, \left[-\frac{N-1}{2}\right]\}$,

Unlike [34], we make the updating of target regression model more robust and $R_{tar}$is updated using Eq. 3.4 if it satisfies the condition $max(\hat{y}_s) > T_a$ & $PSR(\hat{y}_s) > T_{psr}$.

3.2 Support vector machine-based estimator

To increase the robustness of tracking algorithm, a detection module is necessary to recover the target when tracking failure occurs due to long-term occlusion and reentering into the

camera view, erroneous input to model update module or out of camera view movement of an object.

Researchers proposed this model of the online detector and carried out redetection on each frame in [13], [49], [50]. To decrease the computational efficiency, certain threshold is defined to activate the detector. Detector is activated only if the maximum value of response map is less than a predefined threshold [3]. Unlike these two approaches, a support vector-based detector in collaboration with Kalman filter-based predictor is implemented in our approach i.e., the proposed detector is activated if either of the following two conditions are true i.e.

i) $max\ (y_s) < T_r$ *and*

ii) $PSR\ (y_s) < T_{psr}$.

SVM is trained incrementally by considering thick training samples around the estimated position. Binary labels have been assigned with respect to overlap ratio as given in [50]. We assume the training set $\{f_i, c_i | i = 1,2, \dots, N\}$ is given having N number of samples in the frame. f_i is the feature vector of i^{th} sample and c_i is the binary class label for i^{th} sample i.e., $c_i \in \{+1, -1\}$. SVM classifier is defined as follows:

$$\min_h \frac{\lambda}{2}||h||^2 + \frac{1}{N}\sum_i l\big(h; (v_i, c_i)\big), \tag{3.8}$$

where h is hyperplane of SVM detector, $l\big(h; (v, c)\big) = max\{0,1, \dots, c\langle h, v\rangle\}$ and $\langle h, v\rangle$ is inner product between v and h. Passive-aggressive algorithm is applied to update the hyperplane parameters as follows:

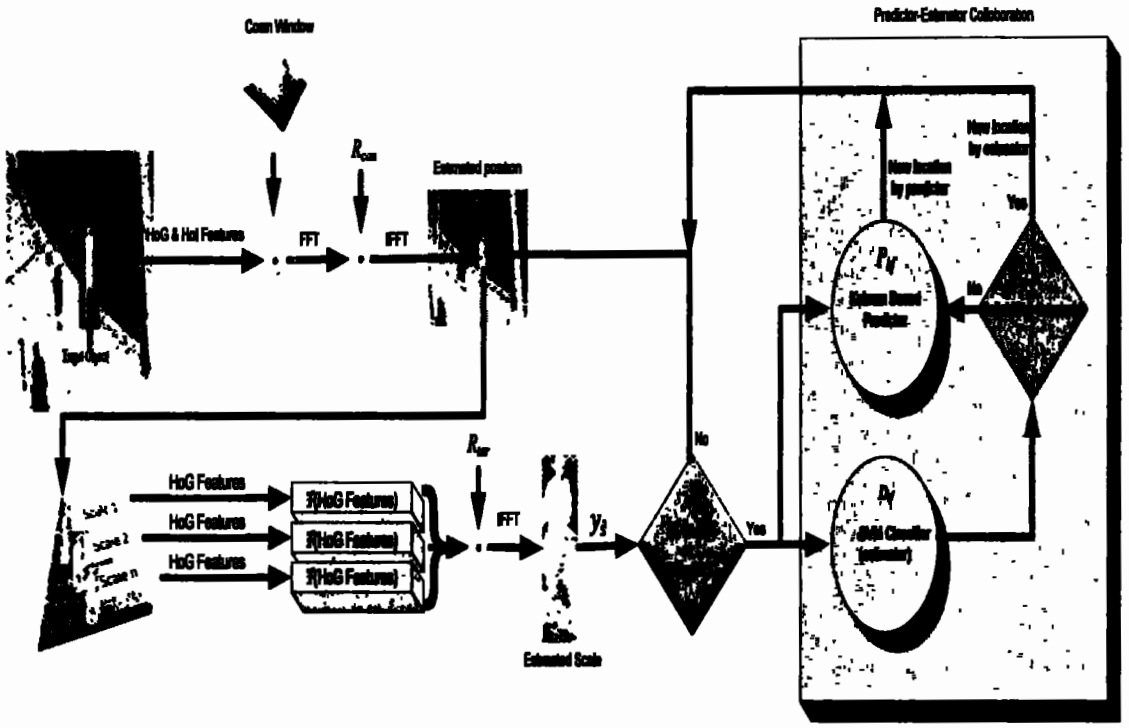

Figure 3.2 Pictorial representation of proposed tracking technique. The Position is estimated, on the estimated position we have estimated the scale using pyramid technique. Reliability of tracking is jugged using conditional block at the right of scale estimation. Result is passed to collaborator if the tracking result is not reliable. SVM classifier estimate the new position and its reliability is checked using second conditional block shown in collaborator module, if the results are reliable enough based on the threshold then new position is SVM estimator based otherwise Kalman filter predictor will give new position. where P_{kf} represents the predictor and D_{rf} represents the detector. $y_{\hat{s}}$ is estimated position at estimated scale, R_{con} is context regression model, R_{tar} is target appearance regression model.

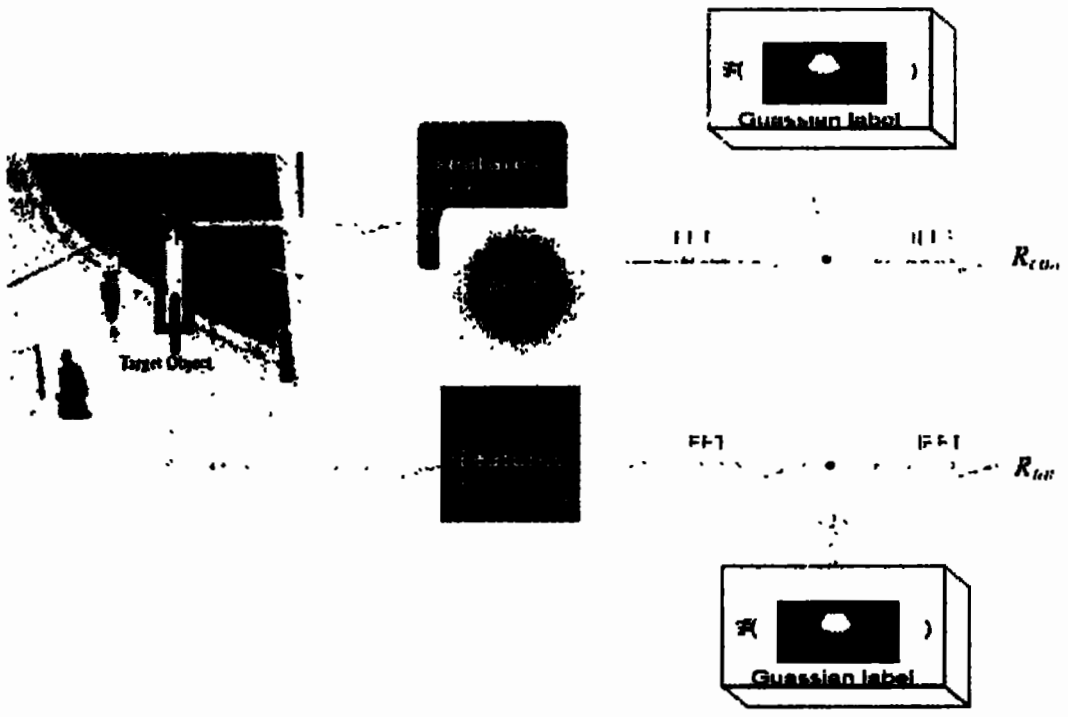

Figure 3.3 Modeling of context regression model and target appearance regression model.

$$h \leftarrow h - \frac{l(h;(\boldsymbol{v},c))}{\|\nabla_h l(h;(\boldsymbol{v},c)\|^2 + \frac{1}{2T}} \nabla_h l\big(h;(\boldsymbol{v},c)\big), \tag{3.9}$$

Where $T \in (0, +\infty)$ controls the rate of updating of h, $\nabla_h l(h;(\boldsymbol{v},c)$ is the gradient of loss function. SVM philosophy is shown in Fig. 3.1.

Unlike existing techniques, in our proposed work the parameters of the detectors are updated using (9), when the max $(y'_{i,t}) > T_i$, where y'_i is the response map value for i^{th} possible state calculated by the detector out of i number of states in X (D_{rf}) for t^{th} frame.

3.3 Kalman filter-based prediction

In literature different Kalman filter [109] based tracking algorithms have been proposed, for example [110], [111], [112], [113]. To increase the efficiency of tracking algorithms, Kalman filter based algorithms have been hybridized with many other tracking algorithms, some of the examples are in [114], [115], [116]. Different from the existing techniques, we incorporated Kalman filter in synchronization with the estimator module to avoid the tracking failure caused by long-term occlusion, motion blur or clutter background. Kalman filter works in a closed loop cycle with prediction and correction steps (10)-(14), respectively. In our proposed tracking framework, Kalman filter is activated in case of the failure of tracking caused by any of the issues mentioned above. During occlusion the Kalman filter takes the current state from the main tracking algorithm (in our case it is KCF) defined by (1)-(4) and predicts the next state by using (10) and (11). The main tracking algorithm (KCF) will stop updating its parameters and target appearance regression model. Kalman filter corrects itself using the previous location predicted by (10) and (11) during occlusion in the next frame. Formulation of Kalman filter is given by Eq. 3.10- Eq. 3.14.

Prediction:

$$x_t'' = \mathrm{A}X''_{t-1} + \mathrm{Bu}, \tag{3.10}$$

$$S_t = AS_{t-1}A^T + Q, \tag{3.11}$$

where, x_t'' is predicted state at t^{th} frame, A is state transition matrix, S_t is posteriori error covariance matrix,Q is covariance matrix of dynamic noise, B is input noise and A is state transition matrix.

Correction:

$$K_{t-1} = S_{t-1}H^T(HS_{t-1}H^T + R)\,, \tag{3.12}$$

$$X''_{t+1} = X_t + K_{t-1}(Y_t - HX''_t)\,, \tag{3.13}$$

$$S_{t+1} = (I - K^T H)\,S_t, \tag{3.14}$$

Where H is measurement matrix, Y_t is the measurement to the Kalman from the main tracking algorithm.

Depending on the condition this measurement may come from either of the three sources i.e.; i) main tracking algorithm ii) estimator module iii) Kalman filter self-prediction in the previous frame. Depending on these two conditions; i) $max\ (y_g) < T_r$ or $PSR\ (y_g) < T_{psr}$ and ii) $max\ (y'_i) > T_t$ Kalman filter continues to predict the next state during occlusion and send predicted state to predictor-estimator collaboration module.

3.4 Prediction estimation collaboration

Collaboration module is proposed to handle long-term occlusion, motion blur and clutter background, unlike existing techniques. Most of the already existing methods model the target using its appearance. The major problem associated with these methods is their incapability of predicting the state of the object during occlusion. When the object re-enters the field of view of frame after occlusion, different tracking techniques have been proposed to recapture the object, such as [3], [14]. Different from existing frameworks, our proposed scheme (AFAM-PEC) activate the predictor and estimator at the same time, when the object gets occluded i.e. $\max(y_g) < T_r$ or $PSR\ (y_g) < T_{psr}$..

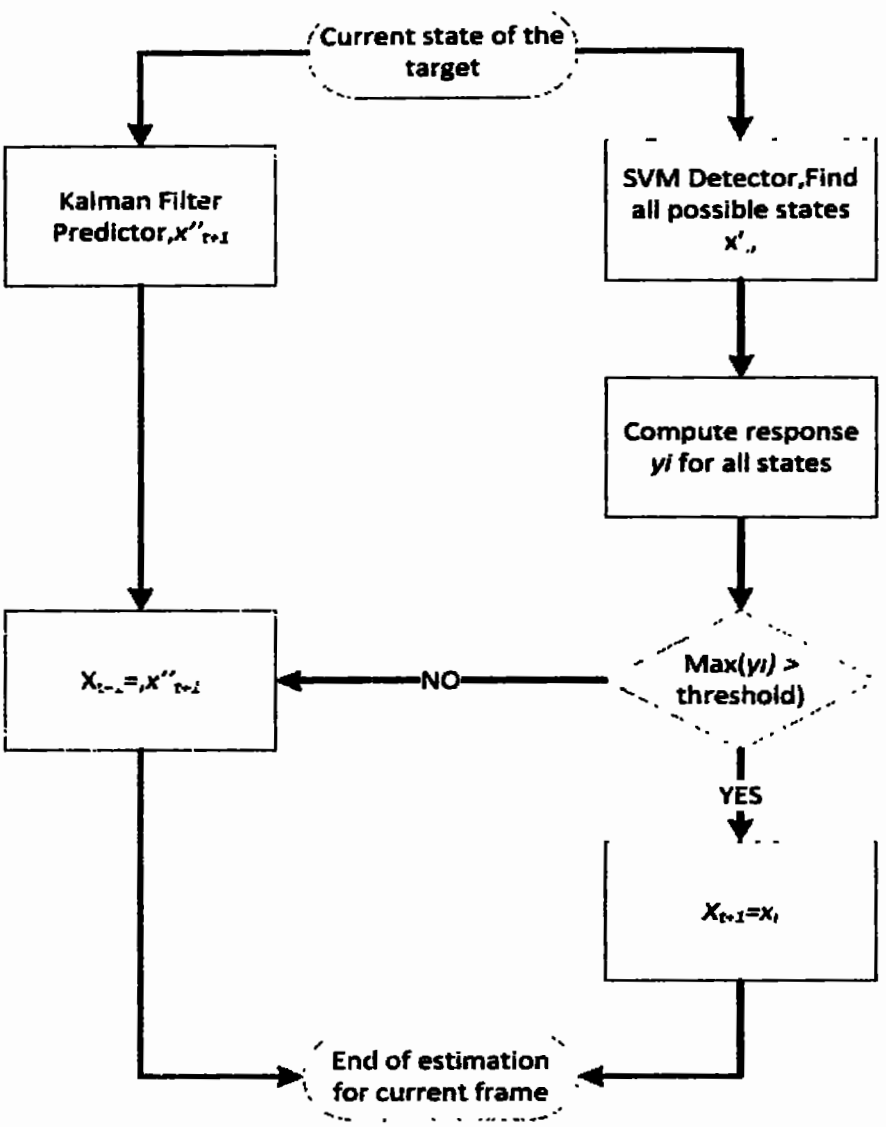

Figure 3.4. Flow chart of Predictor-estimator collaborator module.

During the occlusion period predictor starts predicting the location of the target and SVM based classifier starts estimating the position of the object. If estimated position by SVM based classifier satisfies the condition max (y_i') >T_t, this position is considered a correct estimate and

is given to Kalman filter as measurement to predict the next location. However, if estimated position does not satisfy the condition max (y_t') >T_t, Kalman filter-based predicted position is given to the estimator to estimate the next location in the next frame and the same is given to Kalman filter to predict the next location in the next frame. This approach shows significant improvement in results in comparison to [3], [16], [17]. The flow chart of this module is given in Fig. 3.4.

3.5 Hybridization of average peak correlation energy and confidence of squared response map

Maximum response value has been used widely as reliability measure in tracking algorithms. During occlusion and motion blur etc. response map changes drastically. So, using only maximum response value as reliability measure is not good enough. Another measure i.e., average peak correlation energy(APCE) is presented in[117] given by eq. 3.15.. APCE tells about the degree of fluctuation of response map. If the object undergoes fast motion, the value of APCE will be low.

$$APCE = \frac{|R_{max} - R_{min}|^2}{mean(\sum_{r,c}(R_{r,c} - R_{min})^2)} \quad (3.15)$$

Where, R_{max} , R_{min} denotes the maximum and minimum value of response map respectively. $R_{r,c}$ denotes the rth row and cth column element of response map.

It has been shown practically that if the target apparently appears in the detection scope, there will be sharper peak in response map and the value of APCE will be smaller. However, if the target is occluded, the peak in the response map will be smoother and the

relative value of APCE becomes larger[118]. This problem is solved by squaring the response map and then finding the confidence of squared response map [118].

Peak of the response map is represented in the nominator of eq. 3.16. Whereas denominator represents the mean square value of the response map.

$$CSRM = \frac{\left|R_{max}{}^2 - R_{min}{}^2\right|^2}{\frac{1}{MN}\sum_{r=1}^{M}\sum_{c=1}^{N}\left|R_{r,c}{}^2 - R_{min}{}^2\right|^2} \tag{3.16}$$

Where, R_{max} , R_{min} denotes the maximum and minimum value of response map respectively. $R_{r,c}$ denotes the rth row and cth column element of response map. M*N is the dimension of response map. We increased the robustness of reliability measure by considering both i.e., APCE and SCRM in the following manner.

$$APCE^i || CSRM^i > Threshold; Target\ is\ raliable$$

Where, $APCE^i$ and $CSRM^i$ denotes the average peak correlation energy and confidence of squared response map for i^{th} frame respectively.

3.6 Novel interpretation of difference of peak correlation

Occlusion detection is one of the biggest challenges to object tracking community. Relevant literature is already discussed in chapter number 2. In this section simplest and novel occlusion detection mechanism is discussed. If the peak correlation response value changes, this means that object is losing its actual presentation. This may happen because of occlusion, motion blur or deformation. Let us suppose PCT and PCT-1 are the peak correlation values current and previous frame respectively. Computing the difference of these two peak correlation values gives us insight bout the tracking reliability.

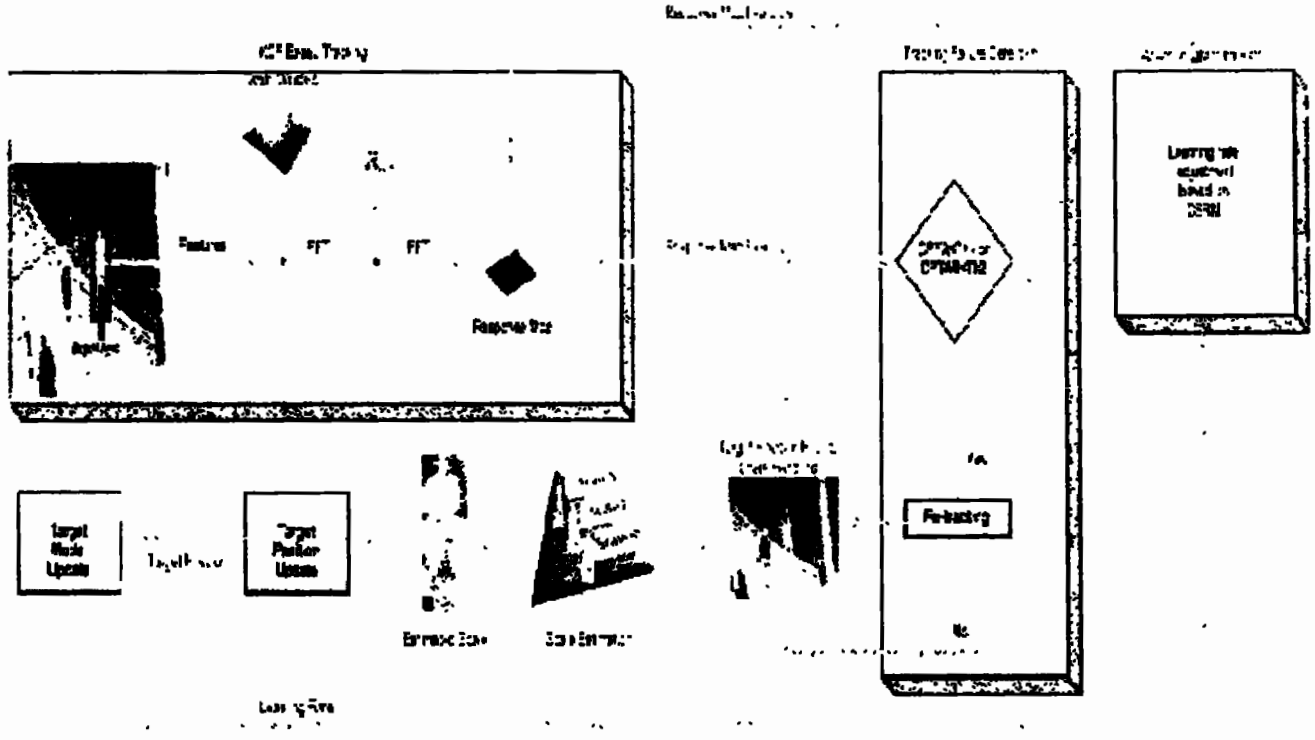

Figure 3.5 Proposed graphical abstract for modified KCF algorithm. Baseline tracker i.e., KCF will give the response map. Multiple features/cues from this map are extracted and fed to next block for occlusion and/or tracking failure is detection. Similarly learning rate is also adjusted based on these multiple cues in the right most block. After getting the reliable results scale is adjusted using scale search algorithm.

Mathematically it is given by eq. 3.17.

$$\begin{cases} if\ (PC_T - PC_{T-1} > 0), Object\ is\ getting\ corrupted \\ if\ (PC_T - PC_{T-1} < 0), object\ is\ comming\ out\ of\ degradaation \\ (PC_T - PC_{T-1}) < Threshold, reliable\ to\ track \end{cases} \quad (3.17)$$

Although peak correlation is extensively used in object tracking algorithms, it is interpreted differently in this study. If the peak correlation of the current frame is less than the peak correlation of previous frame, the object is losing its description. Many factors are responsible for decline in the value of peak correlation some of them are occlusion, motion blur and deformation in object of interest. Whereas if the peak correlation of successive frames is higher than the previous frame this tells us that result is reliable and object is coming out of occlusion and attaining its original description. This strategy is incorporated in the graphical abstract shown in Fig. 3.5.

3.7 Implementation details

The complete flow of the proposed tracking scheme is presented in Algorithm 1. The corresponding flow chart of the novel detector-estimator collaboration module is presented in Fig. 3.4.

This thesis computed multilayer features at fraction of cost using the technique presented in [53]. Histogram of the oriented gradient with 31 bins in histogram along with histogram of local intensities (HoI) with 6×6 windows using 8 bins is implemented. To cater fast illumination variations, HoI is applied on brightness channel and transformed brightness channel as given in [119] is implemented.

Context regression model R_{con} is trained using forty-seven channels feature vector. Whereas target appearance regression model R_{tar} is trained using HoG features with 31 number of bins

only. Constant velocity model of Kalman filter is implemented. The gaussian kernel is used in both target appearance regression model and context-aware regression model. Correlation in (2) and (3) is computed in the Fourier domain. Detection is done by sliding window scanning fashion similar to [36]. SVM classifier is trained considering very large number of samples around the estimated location. Samples having an overlap ratio with the target model bounding box greater than 50% are given positive labels, whereas samples having an overlap ratio less than 10% are assigned negative labels. Regularization parameter in (1) is assumed to be 10^{-4}, search window in frame is 180% of the target object size, width of kernel is set 0.1, learning rate β is considered 0.01. For scale handling, 21 number of scales are considered and α scale factor is considered 1.08. To turn on the SVM-based detector and Kalman filter-based predictor, threshold T_r is considered 0.25. Detectors results are considered reliable only if the threshold $T_t > 0.5$. The second threshold T_t is considered 0.5. Threshold T_a for updating of target regression model R_{tar} is 0.5. Most of the parameters are based on [3], [14], with slight variation or no variation at all. The proposed tracking scheme is implemented in MATLAB (2019) on intel core i7, 7th generation, 2.80 GHz processor, RAM 16GB, a machine with 64bit windows 10 operating system.

Chapter 4.

Results and Discussion

This chapter first describes the dataset and its attributes. Detailed discussion on qualitative and quantitative results is also presented.

We evaluate our tracker quantitively using; i) Distance precision metric as per Fig. 4.1 b, Fig. 4.1 c, Fig. 4.2 b, Fig 4.2 c, Fig. 4.3 b, Fig. 4.3 c and Table 4.3, ii) Overlap success rate metric as per Fig. 4.6, Fig. 4.7, Fig. 4.8, Fig. 4.9, and Table 4.4. Processing time comparison is also given in Table 4.5. We compared our tracker on benchmark dataset videos with long-term correlation tracker (LSTM) [3], spatio-temporal context learning (STC)[16], and real time compressive tracking (CT)[17].

4.1 Dataset

Our proposed tracking scheme is evaluated and compared on number of selected videos form benchmark datasets OTB50 [120], OTB100 [121], TColor-128 [122], and UAV-128 [123]. OTB50 contains 50 videos, OTB100 contains 100 videos, TColor contains 128 color sequence, and UAV-128 contains 128 videos, captured using unmanned air vehicle. Each video has one or more object tracking challenges associated with it. We choose the videos having seven attributes namely, i) occlusion ii) scale variation iii) motion blur iv) fast motion v) out-of-plane rotation vi) deformation and vii) background clutter to support and evaluate our proposed tracker. Explanation of each attribute is given in Table 4.2.

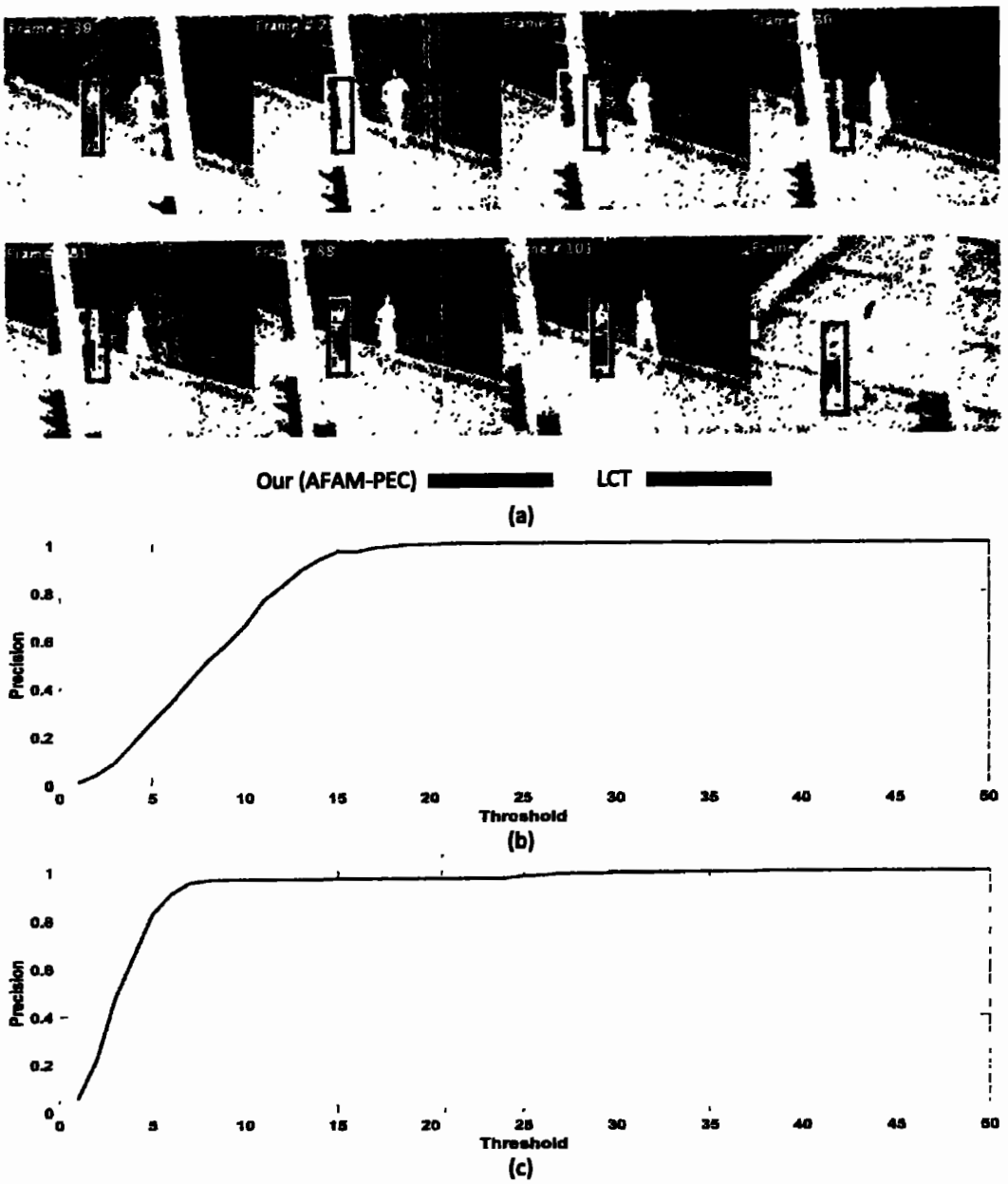

FIGURE 4.1. Comparison of LSTM [3] and proposed algorithm (AFAM-PEC) for Jogging1 video. 4.1(a) Qualitative Analysis. 4:1(b) Distance Precision Plot; Proposed algorithm (AFAM-PEC) achieved distance precision of 100% at threshold of 20-pixels. 4.1(c) Distance Precision Plot; LSTM [3] is unable to achieve 100% distance precision at threshold of 20-pxels.

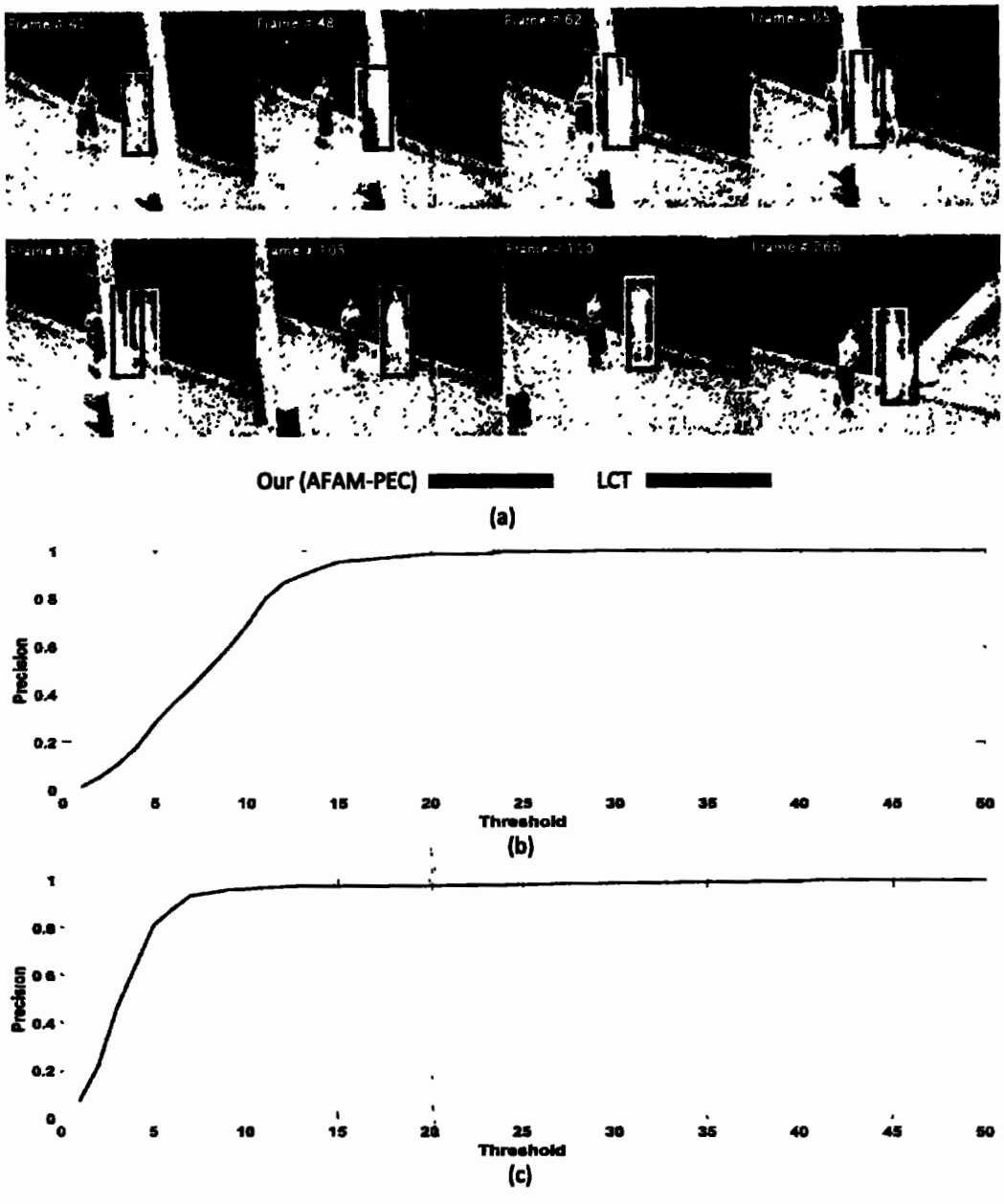

Figure 4.2 Comparison of LSTM [3] and proposed algorithm (AFAM-PEC) for Jogging2 video. 4.2(a) Qualitative Analysis. 4.2(b) Distance Precision Plot; Proposed algorithm (AFAM-PEC) achieved distance precision of 100% at threshold of 20-pixels. 4.2(c) Distance Precision Plot; LSTM [3] cannot achieve 100% distance precision at threshold of 20-pxels.

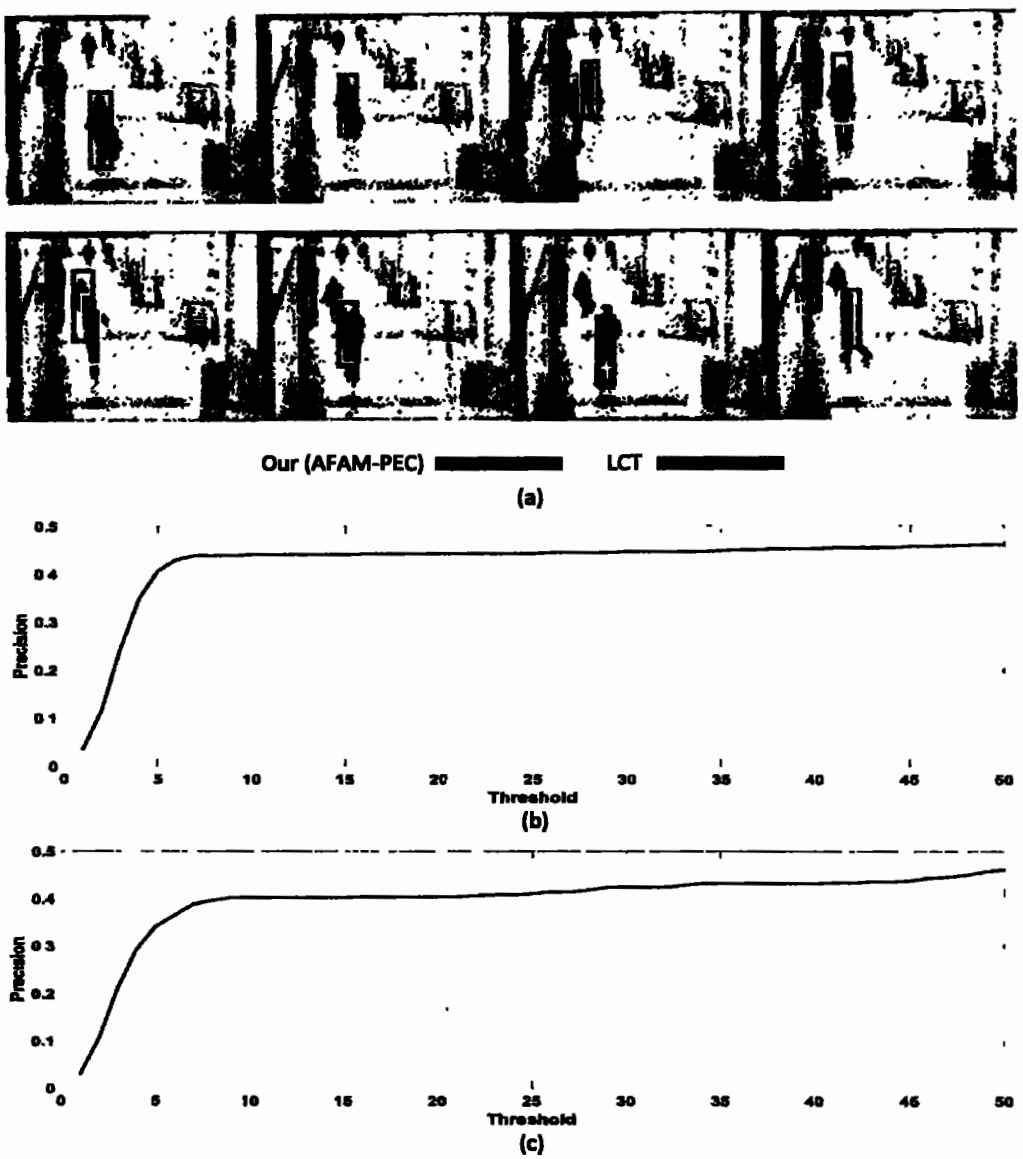

Figure 4.3 Results without predictor-estimator collaboration. 4.3(a) Shows qualitative analysis i.e., after the occlusion in frame number 234, LSTM [3] misguides the Kalman filter-based tracker. Kalman filter-based tracker starts following LSTM [3] and predicting false position.4.3(b) Distance precision plot for Walking2 sequence using Kalman filter-based tracker taking a measurement from LSTM [3]. 4.3(c) Distance precision plot for Walking2 sequence using LSTM [3] without incorporating Kalman filter-based tracker.

4.1.1 Attributes of dataset

This paragraph explains the attributes of each video which also known as challenging aspects by visual tracking community. Six videos shown in Fig. 4.4 i.e. Jogging1, Jogging2, Walking2, Human3, Girl2, Skating2 are selected from OTB100 dataset. These six videos are also part of the TColor-128 dataset. Whereas Fig. 4.5 shows five videos i.e. Bike3, Car4, Car9, Busstation and Building3. Out of these five videos, four videos are part of UAV-123 Dataset and single video Busstation is from the TColor-128 dataset. Hence total of six videos are from OTB100, seven video sequences from TColor-128 and four video sequences from UAV-123 are used to evaluate our propose AFAM-PEC tracker. Jogging1 and Jogging2 sequences have occlusion, deformation and out-of-plane rotation. Walking2 sequence has attributes of scale variation, occlusion and low resolution. Girl2 and Human3 video sequence have maximum challenges i.e. 5. Challenges associated with Girl2 video sequence are namely, scale variation, occlusion, deformation, motion blur and out of plane rotation. Whereas Human3 video contains scale variation, occlusion, deformation, out-of-plane rotation and background clutter. Skating2 sequence has four attributes associated with it i.e., scale variation, occlusion, fast motion and out of the plane rotation. Bike3 contains fast motion, occlusion and out-of-Plane rotation. Car4 and Car9 has Occlusion and Scale Variation. Bustation video sequence has Clutter Background and Occlusion. Finally, Building3 video sequence contains Out of the Plane Rotation. Therefore, a total of seven attributes are associated with these selected eleven videos. Each attribute is explained in Table 4.2.

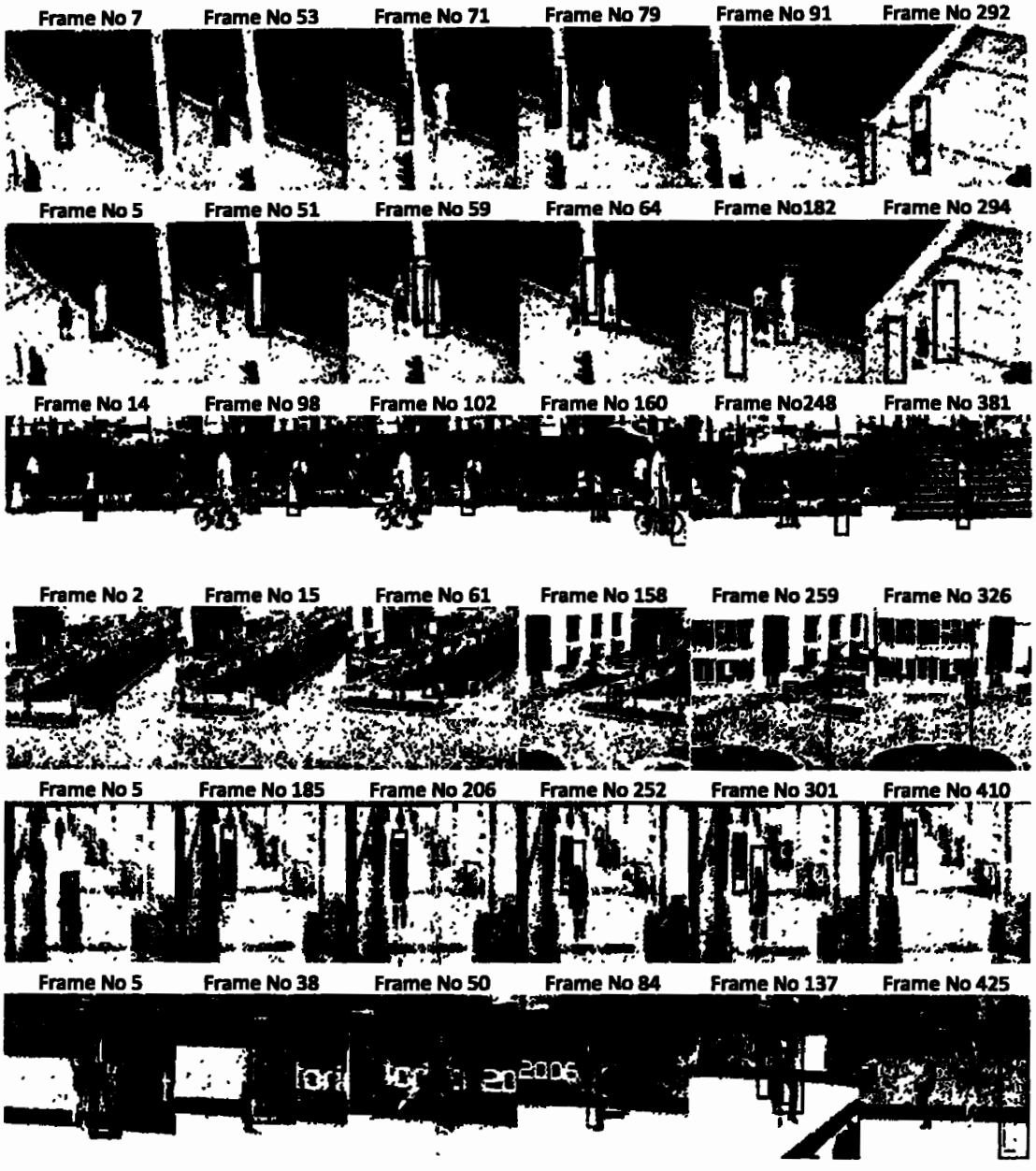

Figure. 4.4 Qualitative results of the proposed scheme (AFAM-PEC), LSTM [3], STC [16], and CT [17] on six challenging sequences selected from OTB50, OTB100 and TColor-128. First row to the last row: jogging1, jogging2, girl2, human3, walking2 and skating2 video sequences are presented respectively.

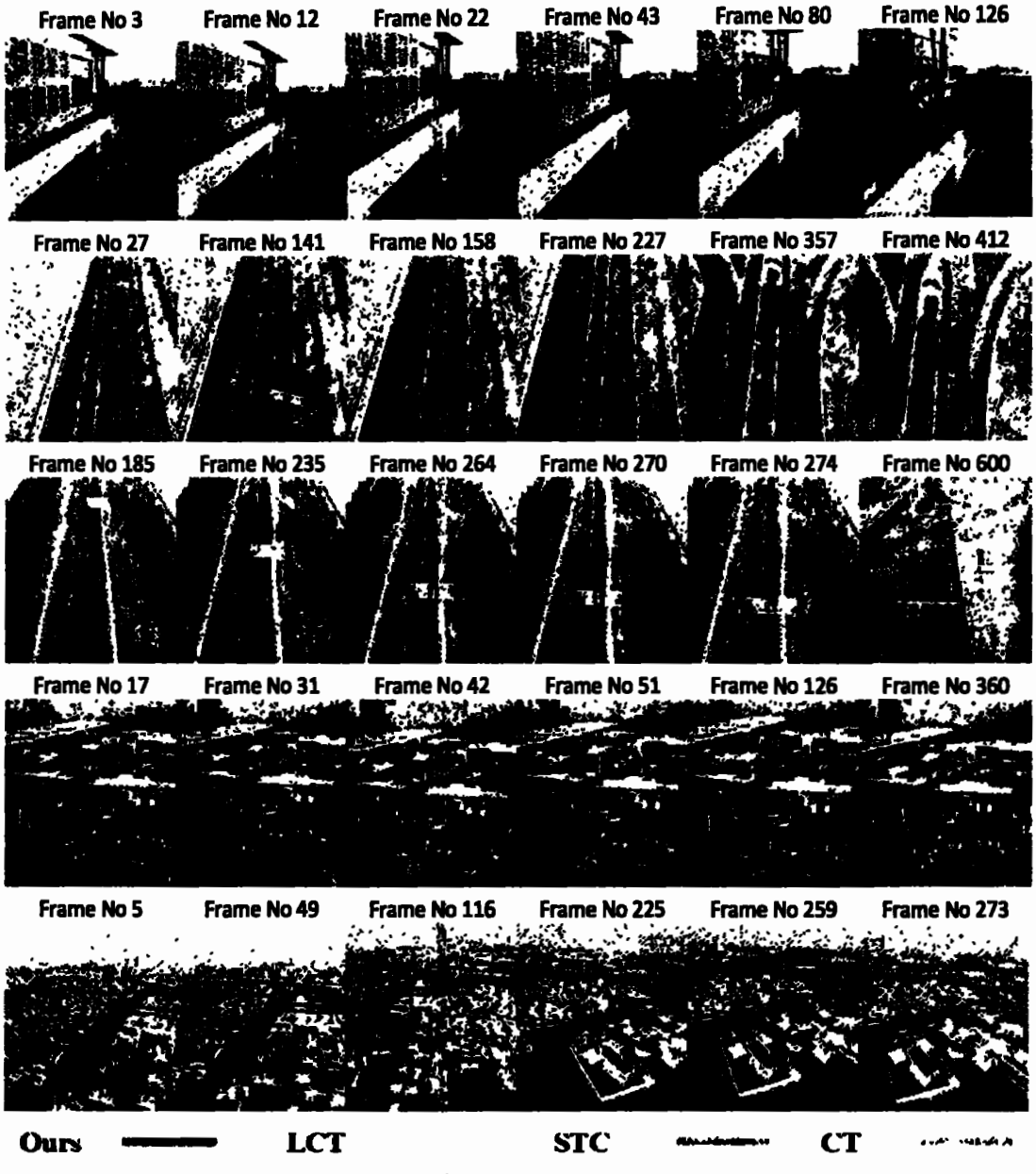

Figure. 4.5 Qualitative results of the proposed scheme (AFAM-PEC), LSTM [3], STC [16], and CT [17] on five challenging sequences selected from TCOLOR-128 and UAV-123 databases. First row to the last row: bike3, car4, car9, busstation, and building3 video sequences are presented respectively.

4.2 Quantitative analysis

Quantitative results in Table 4.3 and Table 4.4 show that our tracker perform well for long-term occlusion challenge. For Jogging2 sequence, the proposed tracker gives distance precision of 100% at 20-pixel threshold and LSTM [3] tracker gives the second-highest precision of 97%. Whereas all the remaining trackers fail to track the object after occlusion. Likewise, on Jogging1 sequence,the proposed scheme again achieves 99% precision.

On Walking2 sequence the proposed tracking scheme again gives 100% distance precision. In contrast, all the remaining three trackers lose the target when the girl in the video sequence gets occluded with boy at frame number 202.

On Girl2 sequence, the proposed tracking algorithm again achieves good performance with a distance precision of 95% and all the other trackers fail to track the target object. On Human3 video sequence our tracker outperforms all the remaining three trackers by achieving distance precision of 99%. It is also worth mentioning that human3 video contains five challenging attributes out of a total of nine attributes given in [121].

Skating2 video sequence has extra challenging attribute of fast motion. Though the proposed tracker and all the other trackers fail to track the target object, our tracker still achieves the second-highest distance precision and tracks the target object for a greater number of frames than LSTM [3]. On this sequence CT [17] tracker tracks the target object more than any other tracker.

Bustation3 video sequence selected from TColor-128 dataset contains severe occlusion and cluttering. Hence all the tracker loses the target very early while the proposed tracking scheme AFAM-PEC achieves the distance precision of 100%. On Bike3 video sequence all the trackers

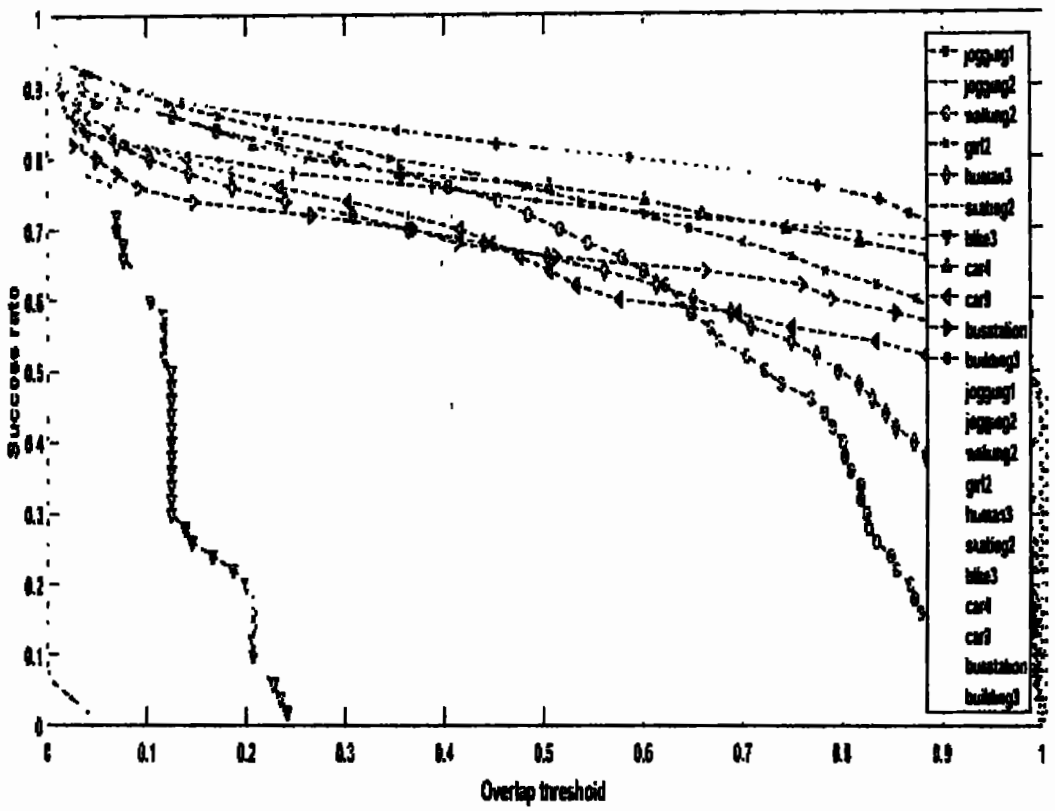

Figure 4.6 Overlap success rate comparison of proposed algorithm (AFAM-PEC) with LSTM [3], Black lines represent the (AFAM-PEC), while yellow lines represent LSTM [3]. The proposed scheme i.e., AFAM-PEC clearly outperform all the other tracker on all the videos except skating2

fail to track the target. The proposed AFAM-PEC achieves the highest precision of 38% among all. On Car4 video sequence again our proposed tacking scheme achieves the 100% distance precision outperforming all other trackers. Similarly, on Car9 video sequence our proposed tracker and LSTM [3] both gave the distance precision of 98% but CT [17] and STC [16] lost the target and gave the precision of less than 25%. Building3 sequence is relatively simpler

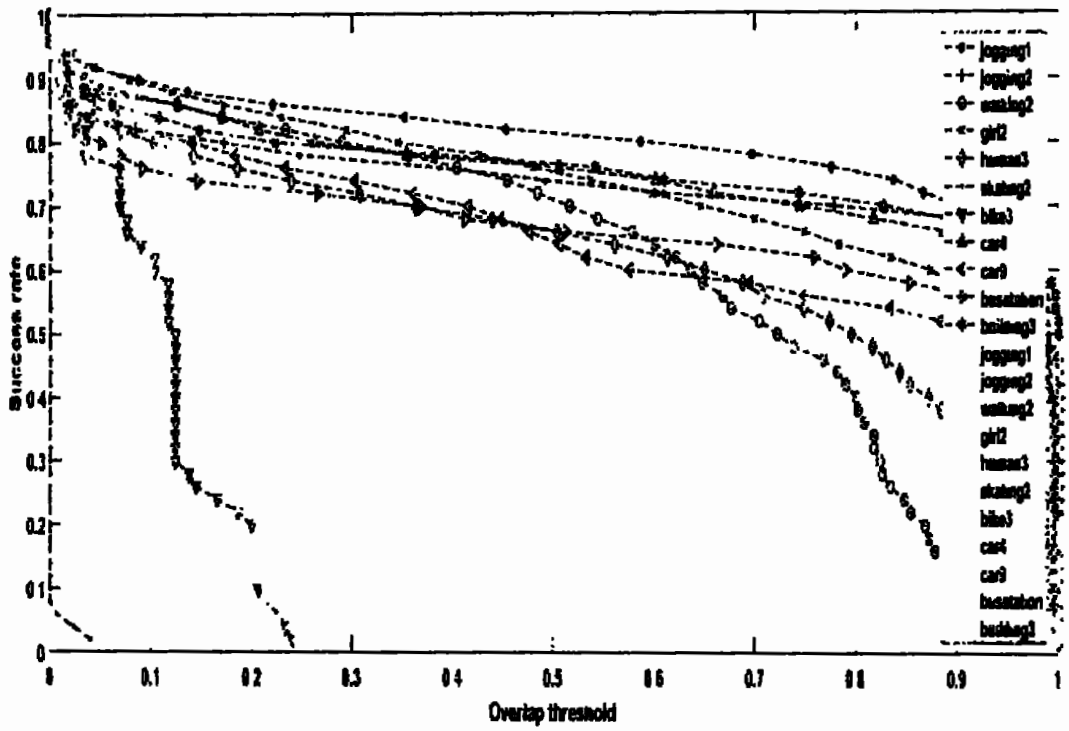

Figure 4.7 Overlap success rate comparison of proposed algorithm (AFAM-PEC) with STC [16]. Black lines represent the (AFAM-PEC), while green lines represent STC [16]. The proposed scheme i.e., AFAM-PEC clearly outperforms all the other trackers on all the videos except skating2.

without having much challenging aspects. So, all the trackers successfully track the target by achieving 100% distance precision. Mean distance precision is also given in Table 4.3.

Our proposed AFAM-PEC achieves the highest mean distance precision of 85%, LSTM [3] achieves the second highest mean precision of 54%, STC [16] achieves the third highest mean distance precision of 38% while CT [17] with lowest mean distance precision of 26%.

We first implemented Kalman filter-based tracking algorithm, as Kalman filter is a measurement follower algorithm. The output of the LSTM [3] algorithm is given as measurement

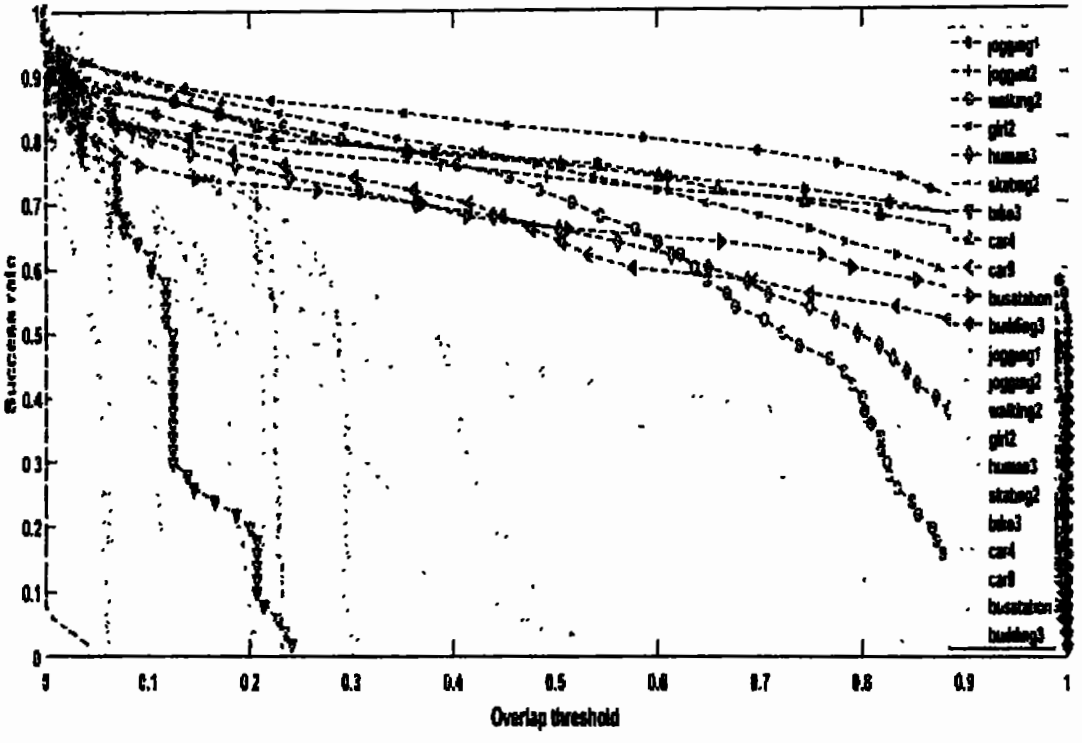

Figure 4.8 Overlap success rate comparison of proposed algorithm (AFAM-PEC) with CT [17]. Black lines represent the (AFAM-PEC), while red lines represent CT [17]. The proposed scheme i.e., AFAM-PEC outperforms all the other trackers on all the videos except skating2.

to Kalman to see the results. Fig. 4.1 shows the results for LSTM [3] and Kalman based tracking algorithm. Kalman filter based tracking algorithm achieves distance precision of almost 100% but the LSTM [3] and most of the traditional algorithms achieves less than 100%. This is because when object gets occluded, LSTM [3] stop estimating correct position of the object and when the object comes out of occlusion LSTM [3] tracker re-detects the target object as shown in Fig. 4.1.

In our implementation Kalman based tracker continuously predicts the new state of the target object even during occlusion which increases the distance precision. To further investigate this

behavior, these algorithms have been applied to other videos and the results are shown in Fig. 4.2 along with the distance precision plot. Frame number 62, 65 and 67 observe this behavior. Fig. 4.3 (Walking2 video sequence) represents another interesting fact that if the measurement (by baseline tracker in our case LSTM [3]) given to Kalman filter is wrong then Kalman will predict the false state in next frame as it is a measurement follower. Now, suppose the baseline tracker continues to give the wrong measurement to Kalman filter-based tracker even after the occlusion of the target is over. In that case, Kalman filter will be predicting the false states and target will be lost.

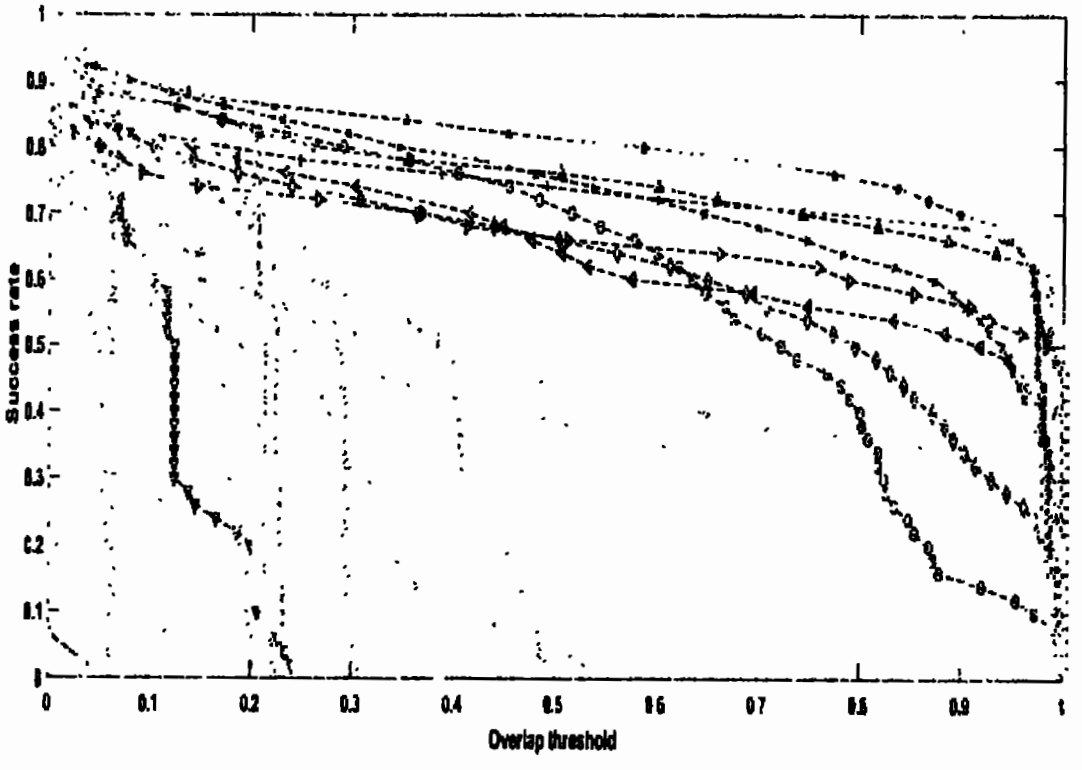

Figure 4.9. Overlap success rate comparison of the proposed algorithm (AFAM-PEC), LSTM [3], STC [16] and CT [17]. Black lines represent the (AFAM-PEC), whereas yellow green and red lines represent the LSTM [3], STC [16] and CT [17] respectively.

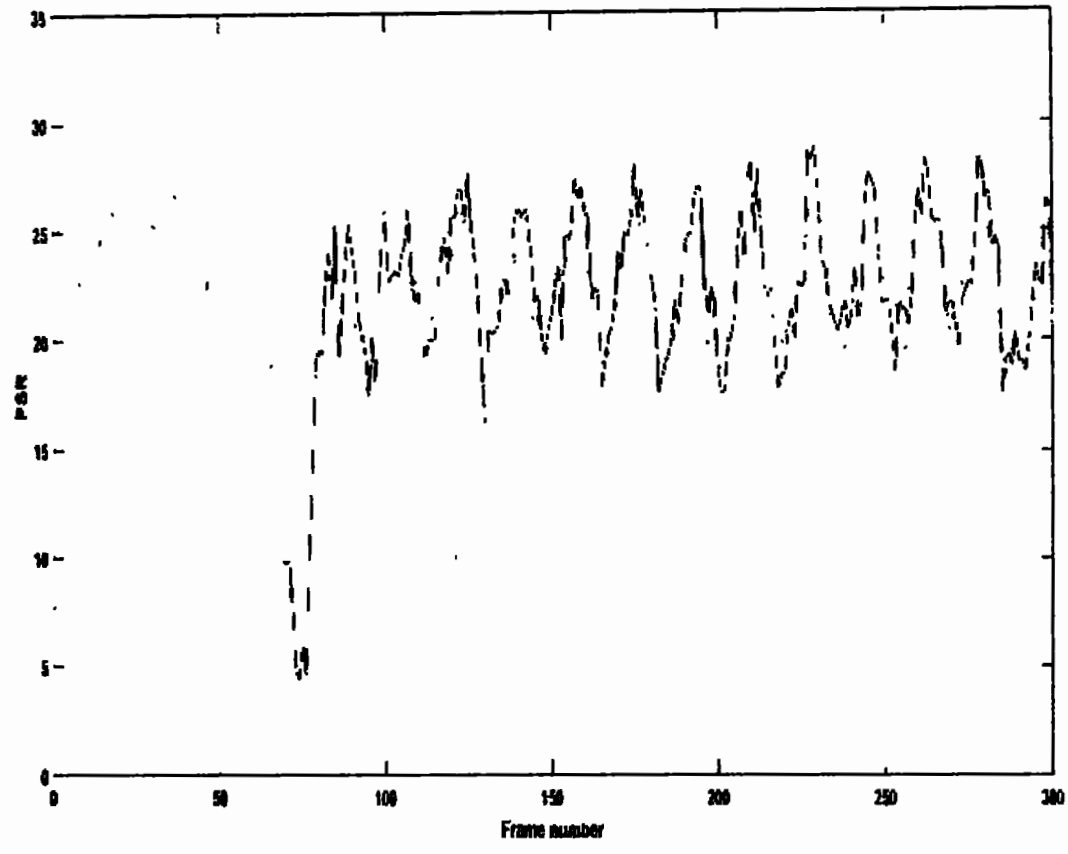

Figure 4.10. Peak correlation score of AFAM-PEC and LSTM [3] for jogging 2 video using blue and red color respectively

This phenomenon is clearly represented in Fig. 4.3, where after 234th frame, baseline tracker misguides the Kalman filter and both the algorithm starts following wrong object. This behavior is corrected by proposing the algorithm which works by using the collaboration of the predictor and estimator. Table 4.3 shows the distance precision of 100% over this video sequence. To further strengthen our argument, Fig. 11 presents the peak to side lobe ratio and. This figure depicts that the proposed tracking algorithm achieves a higher PSR earlier than the LSTM [3]

algorithm. Distance precision plot of Jogging1, Jogging2 and Walking2 sequences are given in Fig. 4.1, Fig. 4.2, and Fig. 4.3 respectively, which shows that the proposed tracking scheme outperforms the other three tracking algorithms.

Before qualitative analysis, let us analyze the overlap success rate metric. Fig. 4.6, Fig. 4.7 and Fig. 4.8 shows the comparison of the proposed algorithm with each of LSTM [3], STC [16], and CT [17]. Whereas Fig 4.9 gives the comparison of all four tracking algorithms on single plot. Table 4.4 shows the overlap success rate for various video sequences at threshold of 0.5. For Jogging1 video LSTM [3] and proposed algorithm gives almost equal success rate of 97%. CT [17] and STC [16] give a 20% and 22% success rate as they fail to track after occlusion. The algorithm proposed in this work achieves the success rate of 99% for Jogging2 video. At the same time, LSTM [3] achieves 97% while other two algorithms achieve less the 20%. Similarly, on all the other remaining video sequences AFAM-PEC achieves the highest success rate, details are given in Table 4.4. The mean overlap success rate is also calculated. The proposed AFAM-PEC outperforms the other trackers by achieving mean overlap success rate of over 75%. LSTM [3] achieves the mean success rate of 51%, STC [16] achieves 30%, and CT [17] achieves 20%.

Table 4.5 shows the frames processed per second (FPS) time of all the 4 trackers over eleven selected videos. Proposed tracking scheme AFAM-PEC shows not much increase in computational cost if compared with the increase in tracking efficiency as per Table 4.3 and table 4.4. For example, our proposed AFAM-PEC process 27.89 frames per second whereas LSTM [3] process 28.30 frames per second. It is difference of even less than one frame. Similar is the case on all the other videos. CT [17] and STC [16] loses the target in most of the video sequences, that is why table 5 shows high FPS for CT [17] and STC [16].

Table 4-1 Notations/variables explanation

Denotation	Symbol	Note
Estimated position and scale	$x_t = (\hat{x}_t, \hat{y}_t, \hat{s}_t)$,	$\hat{x}_t$, $\hat{y}_t$, position of the object and $\hat{s}_t$ is estimated scale
Correlation response map	y_t	At t^{th} frame
Regression model	R_{con}	With respect to context
Regression model	R_{tar}	Model of target
Detector module	D_{svm}	SVM based
Predictor module	P_{kf}	Kalman based
Estimated new position	$\hat{y}_t$	At t^{th} frame
Predicted state	x_t''	State by P_{kf} at t^{th} frame
Estimated states	$X(D_{rf})$	All possible states by detector
Estimated state	x'_i	Estimated possible state I for t^{th} frame
Response map value	y'_i	Response map value for estimated state i

TABLE 4-2 Challenging aspects for visual object tracking

Attribute name	Abbreviation	Explanation
Occlusion	OCC	Target is partially or fully hide behind another object
Scale variation	SV	Bounding boxes ratio of initial frame and present frame is out of range ts, ts > 1 (ts=2).
Out-of-plane rotation	OPR	Rotation of target object out of image plane.
Motion blur	MB	Blurring of target region due to motion.
Fast motion	FM	Ground truth motion is greater than 20 pixels.
Deformation	DEF	Non-rigid object deformation.
Background clutters	BC	Target object background having similar color or texture as that of the target.

4.3 Qualitative analysis

For the qualitative analysis, the results of four trackers i.e. this paper (AFAM-PEC), LSTM [3], STC [16] and CT [125], over eleven video sequences are presented in Fig. 4.4 and Fig. 4.5. The top to bottom rows of Fig. 4.4 contains Jogging1, Jogging2, Girl2, Human3, Walking2 and Skating2 sequences. Row-wise analysis is given in this paragraph. In the first row, all the trackers successfully track the object until frame number 71. While in frame number 79, the proposed algorithm is the only one to track the object exactly while all the others have a window on the pole instead of target object. At frame number 91, LSTM [3] tracker successfully redetects the target. After this frame, the proposed algorithm and LSTM [3] successfully tracks the object till the end of video while the other two algorithms fail to track after the occurrence of occlusion.

Similarly, in the second row of Fig. 4.4, the proposed algorithm successfully tracks the object and redetect the object after occlusion, earlier than all the remaining three algorithms. LSTM [3] shows second-best behavior over this sequence by tracking the object successfully till the end, whereas remaining two algorithms fail to track the object when it reappears after the occlusion. The only issue with LSTM [3] reported for Jogging1 and Jogging2 is the estimation of the position of object during occlusion. In the third row of Fig. 4.4, for Girl2 video sequence, all the trackers successfully track the object until occurrence of cluttering. It can be seen in frame number 98 after the cluttering that all the trackers are successful in tracking, but CT [17] fails to track. When the second challenge of associated with this video occurs i.e. full occlusion, all the trackers fail after the reappearance of the target object except the proposed tracker. Tracker proposed in this paper successfully tracks the target object after occlusion which is visible in frame number 168. We run all the trackers over this video for 600 frames. The fourth row of Fig.

4.4 contains the images from Human3 sequence, again at this sequence our proposed algorithm shows better results i.e., all the trackers lose the object in the start of sequence, but the proposed tracker successfully tracks the object, which is visible in frame numbers 252, 301 and 410. All the trackers are tested over 590 frames of this sequence.

In the Walking2 sequence, the target object gets occluded at frame number 50 and after this frame, CT [17] and LSTM [3] loses the target and starts following wrong object. STC [16] can still track the object after occlusion but fails to handle the scale properly, whereas our proposed scheme successfully tracks the object keeping the right scale. Last sequence shown in Fig. 4.4 is Skating2. Although not any tracker is able to track the object over full video sequence, but our proposed algorithm tracks the object over a greater number of frames than the state of the art i.e. LSTM [3]. The proposed algorithm tracks the object until 125th frame and after this it starts drifting. Whereas CT [17] tracks the object for the maximum number of frames. It can be seen in frame number 484 that all the trackers lose the target. First, STC [16] loses the target after this LSTM [3] then our proposed tracker and at the end CT [17] tracker loses the target. Although our proposed tracker loses the target before CT [17] in this video sequence, but it has benefit of performing better then CT [17] on all the other videos which contain six challenges as per table 4.2 excluding fast motion. Top to bottom rows of Fig. 4.5 contain Bike3, Car4, Car9, Busstation, and building3 video sequences. These sequences are made using UAV. In Bike3 sequence target is relatively small as compared to other video sequences. In this video not any tracker is able to track the target correctly but still proposed tracker performs better and track the target correctly upto 43rd frame.

Table 4-3 Quantitative analysis; distance precision at threshold of 20 pixels over 11 challenging videos selected from otb50,0tb100, tcolor-128, and uav-123

S. NO.	SEQUENCE	OUR AFAM-PEC	LSTM [3]	STC [16]	CT [17]
1	**Jogging 1 OTB100/TColor-128**	**0.9739**	*0.967*	0.208	0.221
2	**Jogging 2 OTB100/TCOLOR-128**	**0.9902**	*0.970*	0.172	0.166
3	**Walking2 OTB100/TColor-128**	**0.722**	0.406	*0.442*	0.382
4	**GIRL2 OTB100/TCOLOR-128**	**0.940**	0.186	*0.262*	0.188
5	**Human3 OTB100/TColor-123**	**0.795**	0.013	*0.088*	0.050
6	**Busstation_cel_clr TColor-128**	**0.9835**	*0.102*	0.099	*0.102*
8	**Car4 UAV-123**	**0.98**	**0.98**	*0.64*	0.285
9	**CAR9 UAV-123**	**0.917**	*0.85*	0.201	0.212
10	**Building3 UAV-123**	**1.000**	**1.000**	*.963*	0.386
Mean Precision		**0.910**	*0.340*	0.250	0.180

Table 4-4 Quantitative Analysis; overlap success rate at Threshold of 0.5 Pixels Over 11 Challenging Videos selected from OTB50, 0TB100, tcolor-128, and uav-123

S. NO.	Sequence	Our Afam-pec	LSTM [3]	STC [16]	CT [17]
1	**Jogging 1 OTB100/tcolor-128**	**0.997**	*0.970*	0.228 Target lost	0.224 Target lost
2	**Jogging 2 otb100/tcolor-128**	**1.000**	*0.973*	.185 Target lost	0.175 Target lost
3	**Walking2 OTB100/tcolor-128**	**1.000**	0.404 Target lost	*0.794 Scale Issue*	0.436 Target lost
4	**Girl2 otb100/tcolor-128**	**0.947**	0.190 Target lost	*0.270 Target lost*	0.115 Target lost
5	**Human3 OTB100/tcolor-123**	**0.988**	.018 Target lost	*0.100 Target lost*	0.055 Target lost
6	**Skating2 otb100/tcolor-128**	0.070 target lost	0.019 Target lost	*0.090 Target lost*	**.190 Target lost**
7	**Busstation_cel_clr tcolor-128**	**1.000**	*0.113 Target lost*	0.110 target lost	0.108
8	**Bike3 uav-123**	**0.379 Target lost**	0.269 Target lost	*0.275*	0.069
9	**Car4 UAV-123**	**1.000**	*0.997*	0.991	0.294
10	**Car9 uav-123**	**0.985**	*0.982*	0.216	0.212
11	**Building3 UAV-123**	**1.000**	**1.000**	**1.000**	**1.000**
Mean success rate -		**0.850**	*0.540*	0.387	0.261

Table 4-5 Quantitative Analysis; frames per second(fps) of 11 Challenging Videos selected from OTB50, 0TB100, tcolor-128, and uav-123

S. NO.	Sequence	Our Afam-pec	LSTM [3]	STC [16]	CT [17]
1	**Jogging 1 OTB100/TColor-128**	**27.89**	*28.30*	34.37 Target lost	24.73 Target lost
2	**Jogging 2 OTB100/TColor-128**	**21.51**	*22.12*	38.23 Target lost	27.07 Target lost
3	**Walking2 OTB100/TColor-128**	**24.65**	*26.39*	34.68 Scale issue	24.24 Target lost
4	**Girl2 OTB100/TColor-128**	**15.62**	*16.63*	12.50 Target lost	21.96 Target lost
5	**Human3 OTB100/TColor-123**	**21.18**	18.09 Target lost	131.02 Target lost	19.93 Target lost
6	**Skating2 OTB100/TColor-128**	**15.28**	20.71 Target lost	58 Target lost	27.82 Target lost
7	**Busstation_cel_clr TColor-128**	**42.40**	54.55 Target lost	20.67 Target lost	12.60 Target lost
8	**Bike3 UAV-123**	**33.14**	57.47 Target lost	22.83 Target lost	13.03 Target lost
9	**Car4 UAV-123**	**13.01**	*16.75*	23.16	13.08 Target lost
10	**Car9 UAV-123**	**10.44**	12.71 Scale issue	29.23 Target lost	18.18 Target lost
11	**Building3 UAV-123**	*14.60*	16.55	22.89	**13.19**

Table 4-6 Quantitative analysis; distance precision at threshold of 20 pixels [3]

Tracker name	Distance precision at 20-pixels	Overlap success rate at 50% threshold
LSTM-Deep[3]	**87.8**	*79.9*
LSTM [3]	84.8	**81.3**
MUSTer[77]	*86.5*	78.4
MEEM[127]	83.0	69.6
TGPR[57]	74.1	62.2
DSST[65]	64.9	61.6
CSK[12]	65.6	55.9
Struck[46]	70.5	62.8
SCM[128]	47.5	37.3
MIL[44]	60.8	52.1
TLD[45]	56.1	45.7
LSHT[129]	54.5	44.3

In second row of Fig. 4.5 Car4 video sequence is show. In this video our proposed AFAM-PEC and LSTM [3] both track the target successfully till the end of video but proposed AFAM-PEC achieves better overall success rate which can be seen in frame numbers 270 ,274 and 600.

STC [16] also able to track the target till the end of video but CT [125] fails to track the object after the occlusion, which is visible in frame number 235.

In the third row of Fig. 4.5, all the trackers successfully track the car till the occurrence of occlusion in frame number 42. At occlusion LSTM [3] struck while our AFAM-PEC successfully tracks the car even when it is occluded. After the occlusion STC [16] also fails to track correctly as per frame number 270. While proposed AFAM-PEC and LSTM [3] track the object till the end of video sequence. Though visually it seems that both the trackers have similar performance but as per quantitative analysis our AFAM-PEC gives better distance precision and overlap success rate. In second last row of Fig. 4.5 AFAM-PEC outperforms all the other trackers by successfully tracking the target after occlusion in frame number 51. All the remaining three trackers fail to track the object after occlusion in this video sequence. Last row of Fig. 4.5 shows the building3 sequence. All the trackers successfully track the target because of simplicity of the video. Table 4.7 gives the comparison of proposed modified KCF algorithm with spatio-temporal context learning [16], state of the art minimum output sum of squared error [4], motion aware correlation filter [118], scale adaptive kernel correlation filter [15]. It is clear from the mean precision that proposed algorithm shows promising result over selected challenging videos.

Table 4.6 gives the comparison of base paper [3] with eleven state of the art tracker. It is shown that adaptive correlation filter with short term and long-term memory gives the highest distance precision when deep features are being used. Whereas without deep features this tracking scheme gives the second highest precision. In our study we are using hand crafted features instead of deep features. Our proposed tracker performs favorable on challenging sequences as per table 4.4 and table 4.3.

Table 4-7 Quantitative analysis; distance precision at threshold of 20 pixels over 7 challenging videos selected from OTB50

Sequence	Proposed	STC	MOSSE	MACF	SAMF
Blurcar2	**1.000**	*0.990*	0.275	**1.000**	0.291
Blurface	**1.000**	0.629	*0.998*	**1.000**	**1.000**
Car1	**1.000**	*0.275*	0.250	**1.000**	**1.000**
Cardark	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Redteam	**1.000**	*0.798*	**1.000**	**1.000**	**1.000**
Trellis	**1.000**	*0.738*	0.178	**1.000**	**1.000**
Walking	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Mean	**1.000**	0.770	0.670	**1.000**	*0.898*

4.4 Summary

This chapter presented the results of experiments performed to evaluate the performance of the proposed study.

Section. 4.1 described the dataset used to evaluate the performance. OTB 50 and UAV 123, two data sets have been used. Sec. 4.2 presented the quantitative results with the help of Table 4.3, Table 4.4, Table 4.5, and Table 4.7. Graphs were also presented in Fig. 4.2, Fig. 4.3, Fig, 4.6, Fig. 4.7, Fig. 4.8, Fig. 4 9 and Fig. 4.10.Sec. 4.3 described the qualitative results using Fig.4.1, Fig. 4.2, Fig. 4.3, Fig. 4.4, and Fig. 4. 5.

It is shown that proposed tracking scheme achieved superior performance in terms of distance precision and overlap threshold. Correlation filter-based tracking method with incorporation of prediction-estimation collaboration module do not increase considerable computational cost. This is verified with the help of frames per second comparison.

Chapter 5 concludes the study with future recommendations.

Chapter 5.

Conclusion and Future Work

This chapter draws a conclusion based on experimental results. In the end, future research recommendations, possibilities, and gaps are discussed.

5.1 Conclusion

In this study, an adaptive correlation filter-based tracking failure avoidance mechanism is presented. A kernelized correlation filter is used as baseline tracker. Failure avoidance mechanism is proposed and integrated with Kernelized correlation filter. In our proposed scheme we first address the occlusion detection problem by soughing two parameters from the response map i.e., i) peak to side lobe ratio ii) peak correlation value. These two parameters work together to detect the occlusion. Second, we incorporated Kalman filter-based predictor and SVM based estimator to kernelized correlation filter. Third, we proposed collaboration module between predictor and estimator to avoid the tracking failure. We choose videos from the standard three datasets (OTB100, TColor-128, and UAV-123) having six challenging attributes to perform the experiments. With the help of experiments, we show that proposed work performs better against state-of-the-art tracking algorithms in terms of distance precision and overlap threshold. Furthermore, the difference of peak correlation value between two consecutive frame is interpreted differently to detect the occlusion and normal scenario in a video. This interpretation is applied to state-of-the-art algorithm Kernelized correlation filter, which shows promising results.

5.2 Future recommendations

Following are some future recommendations and research directions for visual object tracking system which could help to improve the performance and robustness of system further.

- As predictor is continuously predicting the location of the object whether the object is present in the frame or not. Efficiency of the tracker may be further enhanced by devising criterion to stop prediction if the object is out of view/ occluded for some specified time.
- In the proposed visual object tracking system, one can exploit nonlinear prediction schemes to predict the object's state. As movement of object in a video may or may not be linear, nonlinear predictor would help to increase the efficiency of the algorithm. For example, other nonlinear models of Kalman filter may be explored to predict the nonlinear movement of object.
- Hybridization of linear and nonlinear predictors may also be a good choice to increase the performance of visual tracking algorithm without significant increase in computational cost.
- Features fusion techniques may be used to model the target object more precisely, which will eventually help in increasing the performance of proposed algorithm.
- Finally, hybridization of deep features with hand-crafted features may be a good candidate for future research.

BIBILOGRAPHY

[1] S. Liu, Y. Wu, E. Wei, M. Liu, and Y. Liu, "StoryFlow: Tracking the evolution of stories," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2436–2445, 2013, doi: 10.1109/TVCG.2013.196.

[2] A. Acm Reference Format: Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv*, vol. 38, p. 45, 2006, doi: 10.1145/1177352.1177355.

[3] C. Ma, J.-B. Bin Huang, X. Yang, and M.-H. H. Yang, "Adaptive Correlation Filters with Long-Term and Short-Term Memory for Object Tracking," *Int. J. Comput. Vis.*, vol. 126, no. 8, pp. 771–796, Aug. 2018, doi: 10.1007/s11263-018-1076-4.

[4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2544–2550, doi: 10.1109/CVPR.2010.5539960.

[5] A. Lukežič, T. Vojíř, L. Čehovin, J. Matas, and M. Kristan, "Discriminative Correlation Filter with Channel and Spatial Reliability," *Int. J. Comput. Vis.*, vol. 126, no. 7, pp. 671–688, Nov. 2016, doi: 10.1007/s11263-017-1061-3.

[6] P. M. Raju, D. Mishra, and R. K. S. S. Gorthi, "Detection based long term tracking in correlation filter trackers," *Pattern Recognit. Lett.*, vol. 122, pp. 79–85, May 2019, doi: 10.1016/j.patrec.2019.02.028.

[7] L. Zhang and P. N. Suganthan, "Robust visual tracking via co-trained Kernelized

correlation filters," *Pattern Recognit.*, vol. 69, pp. 82–93, Sep. 2017, doi: 10.1016/j.patcog.2017.04.004.

[8] A. Lukežič, L. Č. Zajc, T. Vojíř, J. Matas, and M. Kristan, "FuCoLoT – A Fully-Correlational Long-Term Tracker," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11362 LNCS, pp. 595–611, doi: 10.1007/978-3-030-20890-5_38.

[9] M. Zhang *et al.*, "Visual Tracking via Spatially Aligned Correlation Filters Network," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11207 LNCS, pp. 484–500, doi: 10.1007/978-3-030-01219-9_29.

[10] X. Xue, Y. Li, and Q. Shen, "Unmanned aerial vehicle object tracking by correlation filter with adaptive appearance model," *Sensors (Switzerland)*, vol. 18, no. 9, Sep. 2018, doi: 10.3390/s18092751.

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010, doi: 10.1109/TPAMI.2009.167.

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7575 LNCS, no. PART 4, pp. 702–715, doi: 10.1007/978-3-642-33765-9_50.

[13] F. Pernici, "FaceHugger: The ALIEN tracker applied to faces," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7585 LNCS, no. PART 3, pp. 597–601, doi: 10.1007/978-3-642-33885-4_61.

[14] C. Ma, X. Yang, C. Zhang, and M. H. Yang, "Long-term correlation tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07-12-June-2015, pp. 5388–5396, doi: 10.1109/CVPR.2015.7299177.

[15] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 8926, pp. 254–265, doi: 10.1007/978-3-319-16181-5_18.

[16] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M. H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8693 LNCS, no. PART 5, pp. 127–141, doi: 10.1007/978-3-319-10602-1_9.

[17] K. Zhang, L. Zhang, and M. H. Yang, "Real-time compressive tracking," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7574 LNCS, no. PART 3, pp. 864–877, doi: 10.1007/978-3-642-33712-3_62.

[18] D. Yuan, X. Lu, D. Li, Y. Liang, and X. Zhang, "Particle filter re-detection for visual tracking via correlation filters," *Multimed. Tools Appl.*, vol. 78, no. 11, pp. 14277–14301, Jun. 2019, doi: 10.1007/s11042-018-6800-0.

[19] D. Yuan, X. Zhang, J. Liu, and D. Li, "A multiple feature fused model for visual object tracking via correlation filters," *Multimed. Tools Appl.*, vol. 78, no. 19, pp. 27271–27290, Oct. 2019, doi: 10.1007/s11042-019-07828-2.

[20] D. Yuan, W. Kang, and Z. He, "Robust visual tracking with correlation filters and metric learning," *Knowledge-Based Syst.*, vol. 195, p. 105697, May 2020, doi: 10.1016/j.knosys.2020.105697.

[21] D. Yuan, N. Fan, and Z. He, "Learning target-focusing convolutional regression model for visual object tracking," *Knowledge-Based Syst.*, 2020, doi: 10.1016/j.knosys.2020.105526.

[22] D. Yuan, X. Li, Z. He, Q. Liu, and S. Lu, "Visual object tracking with adaptive structural convolutional network," *Knowledge-Based Syst.*, 2020, doi: 10.1016/j.knosys.2020.105554.

[23] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," *2012 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1838–1845, 2012.

[24] S. Siena and B. V. K. V. Kumar, "Detecting occlusion from color information to improve visual tracking," in *2016 IEEE International Conference on Acoustics,*

Speech and Signal Processing (ICASSP), 2016, pp. 1110–1114, doi: 10.1109/ICASSP.2016.7471848.

[25] A. S. Montero, J. Lang, and R. Laganière, "Scalable Kernel Correlation Filter with Sparse Feature Integration," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015, pp. 587–594, doi: 10.1109/ICCVW.2015.80.

[26] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, vol. 2, pp. 1150–1157 vol.2, doi: 10.1109/ICCV.1999.790410.

[27] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002, doi: 10.1109/TPAMI.2002.1017623.

[28] J. Sivic and A. Zisserman, "Video Google: a text retrieval approach to object matching in videos," *Proc. Ninth IEEE Int. Conf. Comput. Vis.*, pp. 1470–1477 vol.2, 2003.

[29] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *2006 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 2169–2178, 2006.

[30] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, 2005, vol. 2, pp. 1458-1465 Vol. 2, doi: 10.1109/ICCV.2005.239.

[31] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher Kernel for Large-Scale Image Classification," in *Computer Vision – ECCV 2010*, 2010, pp. 143–156.

[32] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," *2010 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 3304–3311, 2010.

[33] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, vol. 1, pp. 886–893 vol. 1, doi: 10.1109/CVPR.2005.177.

[34] L. Bourdev, S. Maji, T. Brox, and J. Malik, "Detecting People Using Mutually Consistent Poselet Activations," in *Computer Vision – ECCV 2010*, 2010, pp. 168–181.

[35] S. Ullman, M. Vidal-Naquet, and E. Sali, "Visual features of intermediate complexity and their use in classification.," *Nat. Neurosci.*, vol. 5, no. 7, pp. 682–687, Jul. 2002, doi: 10.1038/nn870.

[36] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, doi: 10.1109/cvpr.2001.990517.

[37] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the Circulant Structure of Tracking-by-Detection with Kernels," in *Computer Vision – ECCV 2012*,

2012, pp. 702–715.

[38] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual Tracking: An Experimental Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014.

[39] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, 2011, doi: https://doi.org/10.1016/j.neucom.2011.07.024.

[40] H. Pirsiavash and D. Ramanan, "Steerable part models," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3226–3233, doi: 10.1109/CVPR.2012.6248058.

[41] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8, doi: 10.1109/CVPR.2008.4587583.

[42] A. Vedaldi, M. Blaschko, A. Zisserman, and IEEE, "Learning Equivariant Structured Output SVM Regressors," *2011 IEEE Int. Conf. Comput. Vis.*, pp. 959–966, 2011.

[43] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-SVMs for object detection and beyond," *2011 Int. Conf. Comput. Vis.*, pp. 89–96, 2011.

[44] B. Babenko, M. H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, 2011, doi: 10.1109/TPAMI.2010.226.

[45] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection," 2010.

[46] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *2011 International Conference on Computer Vision*, 2011, pp. 263–270, doi: 10.1109/ICCV.2011.6126251.

[47] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," *2008 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1–8, 2008.

[48] B. Alexe, V. Petrescu, and V. Ferrari, "Exploiting spatial overlap to efficiently compute appearance distances between image windows," in *Advances in Neural Information Processing Systems*, 2011, vol. 24.

[49] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," *CoRR*, vol. abs/1404.7, 2014.

[50] D. S. Bolme, B. A. Draper, and J. R. Beveridge, "Average of Synthetic Exact Filters," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2105–2112, doi: 10.1109/CVPR.2009.5206701.

[51] M. Zhang, J. Xing, J. Gao, and W. Hu, "Robust visual tracking using joint scale-spatial correlation filters," in *Proceedings - International Conference on Image Processing, ICIP*, 2015, vol. 2015-Decem, pp. 1468–1472, doi: 10.1109/ICIP.2015.7351044.

[52] C. Dubout and F. Fleuret, "Exact Acceleration of Linear Object Detectors," in

Computer Vision – ECCV 2012, 2012, pp. 301–311.

[53] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, 2014, doi: 10.1109/TPAMI.2014.2300479.

[54] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 261–271, Feb. 2007, doi: 10.1109/TPAMI.2007.35.

[55] Q. Bai, Z. Wu, S. Sclaroff, M. Betke, and C. Monnier, "Randomized ensemble tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2040–2047, doi: 10.1109/ICCV.2013.255.

[56] W. Li and Y. Lin, "Adaptive Randomized Ensemble Tracking Using Appearance Variation and Occlusion Estimation," 2016, doi: 10.1155/2016/1879489.

[57] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with Gaussian processes regression," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8691 LNCS, no. PART 3, pp. 188–203, doi: 10.1007/978-3-319-10578-9_13.

[58] B. Huang, T. Xu, J. Li, Z. Shen, and Y. Chen, "Transfer learning-based discriminative correlation filter for visual tracking," *Pattern Recognit.*, vol. 100, p. 107157, Apr. 2020, doi: 10.1016/j.patcog.2019.107157.

[59] X. Zhen, S. Fei, Y. Wang, and W. Du, "A Visual Object Tracking Algorithm Based on

Improved TLD," *Algorithms*, vol. 13, no. 1, p. 15, Jan. 2020, doi: 10.3390/a13010015.

[60] B. V. K. Vijaya Kumar, A. Mahalanobis, and R. D. Juday, *Correlation Pattern Recognition*, vol. 9780521571036. Cambridge University Press, 2005.

[61] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015, doi: 10.1109/TPAMI.2014.2345390.

[62] C. Ma, X. Yang, C. Zhang, and M. H. Yang, "Learning a temporally invariant representation for visual tracking," in *Proceedings - International Conference on Image Processing, ICIP*, 2015, vol. 2015-December, pp. 857–861, doi: 10.1109/ICIP.2015.7350921.

[63] M. Danelljan, F. S. Khan, M. Felsberg, and J. Van De Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1090–1097, doi: 10.1109/CVPR.2014.143.

[64] Y. Qi, L. Qin, S. Zhang, Q. Huang, and H. Yao, "Robust visual tracking via scale-and-state-awareness," *Neurocomputing*, vol. 329, pp. 75–85, Feb. 2019, doi: 10.1016/j.neucom.2018.10.035.

[65] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*, 2014, doi: 10.5244/c.28.65.

[66] J. Fan, H. Song, K. Zhang, Q. Liu, F. Yan, and W. Lian, "Real-time manifold regularized context-aware correlation tracking," *Front. Comput. Sci.*, vol. 14, no. 2, pp. 334–348, Apr. 2020, doi: 10.1007/s11704-018-8104-y.

[67] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, vol. 2019-June, pp. 4665–4674, doi: 10.1109/CVPR.2019.00480.

[68] Y. She and Y. Yi, "Learning multi-feature based spatially regularized and scale adaptive correlation filters for visual tracking," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020, vol. 11961 LNCS, pp. 480–491, doi: 10.1007/978-3-030-37731-1_39.

[69] V. N. Boddeti, T. Kanade, and B. V. K. V. Kumar, "Correlation Filters for Object Alignment," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2291–2298, doi: 10.1109/CVPR.2013.297.

[70] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for Correlation Filter based tracking," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-January, pp. 5000–5008, doi: 10.1109/CVPR.2017.531.

[71] A. Rodriguez and B. V. K. V. Kumar, "Dealing with circular correlation effects," in *Automatic Target Recognition XXIII*, 2013, vol. 8744, pp. 190–196.

[72] O. Akin, E. Erdem, A. Erdem, and K. Mikolajczyk, "Deformable part-based tracking by coupled global and local correlation filters," *J. Vis. Commun. Image Represent.*, vol. 38, pp. 763–774, 2016, doi: https://doi.org/10.1016/j.jvcir.2016.04.018.

[73] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," *2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1401–1409, 2016.

[74] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE 1 High-Speed Tracking with Kernelized Correlation Filters."

[75] M. Kristan *et al.*, "The Visual Object Tracking VOT2014 Challenge Results," in *Computer Vision - ECCV 2014 Workshops*, 2015, pp. 191–217.

[76] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning Color Names for Real-World Applications," *Trans. Img. Proc.*, vol. 18, no. 7, pp. 1512–1523, Jul. 2009, doi: 10.1109/TIP.2009.2019809.

[77] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "MUlti-Store Tracker (MUSTer): A cognitive psychology inspired approach to object tracking," *2015 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 749–758, 2015.

[78] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, pp. 84–90, 2012.

[79] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on*

Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[80] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[81] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in *Computer Vision -- ECCV 2020*, 2020, pp. 213–229.

[82] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in *Proceedings of the 37th International Conference on Machine Learning*, 2020, vol. 119, pp. 1597–1607.

[83] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: delving deep into convolutional nets," 2014, pp. 1–12.

[84] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional Features for Correlation Filter Based Visual Tracking," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015, pp. 621–629, doi: 10.1109/ICCVW.2015.84.

[85] L. Wang, W. Ouyang, X. Wang, and H. Lu, "STCT: Sequentially Training Convolutional Networks for Visual Tracking," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1373–1381, doi:

10.1109/CVPR.2016.153.

[86] J. Johnander, M. Danelljan, F. S. Khan, and M. Felsberg, "DCCO: Towards Deformable Continuous Convolution Operators for Visual Tracking," in *Computer Analysis of Images and Patterns*, 2017, pp. 55–67.

[87] H. Zuo, Z. Xu, J. Zhang, and G. Jia, "Visual tracking based on transfer learning of deep salience information," *Opto-Electronic Adv.*, vol. 03, p. 190018, 2020.

[88] M. Kristan *et al.*, "The Visual Object Tracking VOT2016 Challenge Results," in *Computer Vision – ECCV 2016 Workshops*, 2016, pp. 777–823.

[89] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Discriminative Scale Space Tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, p. 1561—1575, Aug. 2017, doi: 10.1109/tpami.2016.2609928.

[90] W. Walid, M. Awais, A. Ahmed, G. Masera, and M. Martina, "Real-time implementation of fast discriminative scale space tracking algorithm," *J. Real-Time Image Process.*, 2021, doi: 10.1007/s11554-021-01119-6.

[91] H. Lu, D. Xiong, J. Xiao, and Z. Zheng, "Robust long-term object tracking with adaptive scale and rotation estimation," *Int. J. Adv. Robot. Syst.*, vol. 17, no. 2, p. 1729881420909736, 2020, doi: 10.1177/1729881420909736.

[92] M. Tang, B. Yu, F. Zhang, and J. Wang, "High-Speed Tracking with Multi-kernel Correlation Filters," 2018, doi: 10.1109/CVPR.2018.00512.

[93] X. An, Q. Liang, and N. Sun, "Multi-kernel support correlation filters with temporal

filtering constraint for object tracking," *Multimed. Tools Appl.*, vol. 80, no. 9, pp. 14041–14073, 2021, doi: 10.1007/s11042-020-10345-2.

[94] X. Li, Q. Liu, Z. He, H. Wang, C. Zhang, and W.-S. Chen, "A multi-view model for visual tracking via correlation filters," *Knowledge-Based Syst.*, vol. 113, pp. 88–99, 2016, doi: https://doi.org/10.1016/j.knosys.2016.09.014.

[95] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *Computer Vision — ECCV'98*, 1998, pp. 484–498.

[96] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4902–4912, doi: 10.1109/CVPR.2015.7299124.

[97] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 1515–1522, doi: 10.1109/ICCV.2009.5459278.

[98] H. Li, Y. Liu, C. Wang, S. Zhang, and X. Cui, "Tracking Algorithm of Multiple Pedestrians Based on Particle Filters in Video Sequences," *Comput. Intell. Neurosci.*, vol. 2016, p. 8163878, 2016, doi: 10.1155/2016/8163878.

[99] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-Backward Error: Automatic Detection of Tracking Failures," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 2756–2759, doi: 10.1109/ICPR.2010.675.

[100] Y. Wu, J. Lim, and M.-H. Yang, "Online Object Tracking: A Benchmark."

[101] M. K. Rapuru, S. Kakanuru, P. M. Venugopal, D. Mishra, and G. R. K. S. Subrahmanyam, "Correlation-Based Tracker-Level Fusion for Robust Visual Tracking," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4832–4842, 2017, doi: 10.1109/TIP.2017.2699791.

[102] H. Nam and B. Han, "Learning Multi-domain Convolutional Neural Networks for Visual Tracking," *2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 4293–4302, 2016.

[103] H. Nam, M. Baek, and B. Han, "Modeling and Propagating CNNs in a Tree Structure for Visual Tracking," *ArXiv*, vol. abs/1608.0, 2016.

[104] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-Convolutional Siamese Networks for Object Tracking," *ArXiv*, vol. abs/1606.0, 2016.

[105] G. Zhu, F. Porikli, and H. Li, "Beyond Local Search: Tracking Objects Everywhere with Instance-Specific Proposals," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 943–951, doi: 10.1109/CVPR.2016.108.

[106] D. Du, H. Qi, L. Wen, Q. Tian, Q. Huang, and S. Lyu, "Geometric Hypergraph Learning for Visual Tracking," *IEEE Trans. Cybern.*, vol. 47, pp. 4182–4195, 2017.

[107] S. Becker, S. B. Krah, W. Hübner, and M. Arens, "MAD for visual tracker fusion," in *Optics and Photonics for Counterterrorism, Crime Fighting, and Defence XII*, 2016, vol. 9995, pp. 166–173.

[108] K. P. Murphy, *(book)Machine Learning: A Probabilistic Perspective*. 1991.

[109] R. E. Kalman, "A new approach to linear filtering and prediction problems," American Society of Mechanical Engineers Digital Collection, Mar. 1960.

[110] Y. Y. Chen, P. H. Chen, and S. C. Chien, "An objective tracking method based on Kalman filter," in *2016 International Conference on Advanced Robotics and Intelligent Systems, ARIS 2016*, 2017, doi: 10.1109/ARIS.2016.7886626.

[111] "(PDF) The Kalman Filter and Related Algorithms: A Literature Review." [Online]. Available: https://www.researchgate.net/publication/236897001_The_Kalman_Filter_and_Related_Algorithms_A_Literature_Review. [Accessed: 25-Mar-2020].

[112] P. Li, T. Zhang, and B. Ma, "Unscented Kalman filter for visual curve tracking," in *Image and Vision Computing*, 2004, vol. 22, no. 2, pp. 157–164, doi: 10.1016/j.imavis.2003.07.004.

[113] Y. Yoon, A. Kosaka, and A. C. Kak, "A new Kalman-filter-based framework for fast and accurate visual tracking of rigid objects," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1238–1251, 2008, doi: 10.1109/TRO.2008.2003281.

[114] A. Ali, A. Jalil, J. Ahmed, M. A. Iftikhar, and M. Hussain, "Correlation, Kalman filter and adaptive fast mean shift based heuristic approach for robust visual tracking," *Signal, Image Video Process.*, vol. 9, no. 7, pp. 1567–1585, Oct. 2015, doi: 10.1007/s11760-014-0612-0.

[115] A. Salhi and A. Y. Jammoussi, "Object tracking system using Camshift, Meanshift and Kalman filter Modeling from an Object and Multi-object Tracking System View project Tuni-Shamballa View project Object tracking system using Camshift, Meanshift and Kalman filter."

[116] V. Karavasilis, C. Nikou, and A. Likas, "Visual tracking by adaptive Kalman filtering and mean shift," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 6040 LNAI, pp. 153–162, doi: 10.1007/978-3-642-12842-4_19.

[117] M. Wang, Y. Liu, and Z. Huang, "Large Margin Object Tracking with Circulant Feature Maps," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4800–4808, doi: 10.1109/CVPR.2017.510.

[118] Y. Zhang, Y. Yang, W. Zhou, L. Shi, and D. Li, "Motion-Aware Correlation Filters for Online Visual Tracking," *Sensors*, vol. 18, no. 11, 2018, doi: 10.3390/s18113937.

[119] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1994, vol. 801 LNCS, pp. 151–158, doi: 10.1007/bfb0028345.

[120] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2411–2418, doi: 10.1109/CVPR.2013.312.

[121] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015, doi: 10.1109/TPAMI.2014.2388226.

[122] P. Liang, E. Blasch, and H. Ling, "Encoding Color Information for Visual Tracking: Algorithms and Benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015, doi: 10.1109/TIP.2015.2482905.

[123] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9905 LNCS, pp. 445–461, doi: 10.1007/978-3-319-46448-0_27.

[124] C. Ma, X. Yang, C. Zhang, and M.-H. H. Yang, "Long-term Correlation Tracking," IEEE Computer Society, Oct. 2015.

[125] K. Zhang, L. Zhang, and M.-H. Yang, "Real-Time Compressive Tracking."

[126] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Adaptive Correlation Filters with Long-Term and Short-Term Memory for Object Tracking," *Int. J. Comput. Vis.*, vol. 126, no. 8, pp. 771–796, 2018, doi: 10.1007/s11263-018-1076-4.

[127] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8694 LNCS, no. PART 6, pp. 188–203, doi: 10.1007/978-3-319-10599-4_13.

[128] W. Zhong, H. Lu, and M.-H. Yang, "Robust Object Tracking via Sparse Collaborative Appearance Model," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2356–2368, 2014, doi: 10.1109/TIP.2014.2313227.

[129] S. He, Q. Yang, R. W. H. Lau, J. Wang, and M.-H. Yang, "Visual Tracking via Locality Sensitive Histograms," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2427–2434, doi: 10.1109/CVPR.2013.314.