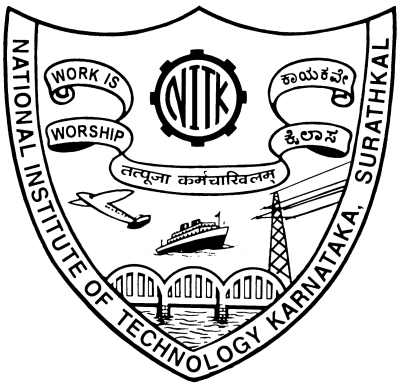
OBJECT TRACKING IN RGB AND INFRARED IMAGERY

Thesis

Submitted in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

by

ASHA C S



DEPARTMENT OF ELECTRONICS AND COMMUNICATION,
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,
SURATHKAL, MANGALORE -575025

AUGUST, 2018

DECLARATION

I hereby ***declare*** that the Research Thesis entitled **OBJECT TRACKING IN RGB AND INFRARED IMAGERY** which is being submitted to the ***National Institute of Technology Karnataka, Surathkal*** in partial fulfillment of the requirements for the award of the Degree of ***Doctor of Philosophy*** is a ***bona fide report of the research work carried out by me***. The material contained in this thesis has not been submitted to any University or Institution for the award of any degree.

ASHA C S
Register No.: EC14F02
Department of Electronics and
Communication Engineering

Place: NITK - Surathkal
Date:

CERTIFICATE

This is to *certify* that the Research Thesis entitled **OBJECT TRACKING IN RGB AND INFRARED IMAGERY**, submitted by **Asha C S** (Register Number: EC14F02) as the record of the research work carried out by her, is *accepted* as the *Research Thesis submission* in partial fulfillment of the requirements for the award of degree of ***Doctor of Philosophy***.

Dr. A V Narasimhadhan
Research Guide
Assistant Professor
Department of Electronics and
Communication Engineering
NITK Surathkal - 575025

Chairman - DRPC
(Signature with Date and Seal)

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support of countless people who have extended their assistance in the completion of work.

Foremost, I express my thankfulness to my supervisor. This work would not have been possible without his support. I would like to thank him for his dedication, motivation, and constant guidance all through the research work. He has been helpful in both academic and personal for which I am grateful.

I am grateful to Prof. U. Sripati, Head, Department of Electronics and Communication Engineering for his constant support. I thank Prof. M S Bhat, former HoD for being supportive all the time. I would also thank Dr. Shyamlal, Dr. Aparna for their inputs during the study. My heartfelt thanks to Dr. Ashwini Chaturvedi and Dr. Jidesh, RPAC members for their valuable suggestions. I also thank all the faculty and staff members of E&C department, NITK Surathkal. Specially, I am thankful to Mrs. Roshni for her help during the study.

I also thank research scholars at NITK for the support they have extended. My sincere thanks to Mrs. Shilpa Kamath, Mr. Shreyas Simu, Miss. Anu Shaju and Mr. Nagaraj, who always helped me technically and personally.

Most importantly, I would like to thank my family members for their never ending support and inspiration throughout the course. My special thanks to my husband Laxmi Narasimha and son Vihan for their understanding and patience.

(Asha C S)

This thesis is dedicated to
My Parents

ABSTRACT

Object tracking is the process of locating the object throughout the frames of a video. This thesis explores tracking of an object selected by the user in RGB and infrared imagery using correlation filters. Also, we investigate illumination invariant tracking in RGB videos using median flow tracker. Additionally, we apply the correlation filter based tracker for multi object tracking to count the vehicles.

The correlation filters have been widely used in computer vision for matching, detection, and tracking purposes. The basic principle of correlation filter is to learn from a set of training data to produce desired target data. The correlation filters appeal to the researchers due to its properties such as shift invariance, real-time speed, immunity to noise, and efficiency. In spite of high accuracy, the correlation filter based tracker has room for further improvements. Also, optical flow based tracker attracted tracking community recently through median flow tracking. However, there is a scope for an extension to achieve better accuracy. Thus, in this thesis, few improvements are suggested to the correlation filters for tracking applications in color and infrared imagery.

The performance of a visual tracker is always degraded due to several reasons that include pose, size, appearance, illumination, occlusion, fast motion, blur, moving camera and so on. However, sudden illumination variation causes the median flow tracker to drift resulting in tracking failure. Hence, illumination invariant techniques are studied to expand the median flow tracker for robust visual tracking.

This thesis considers the combination of discriminative and generative techniques by switching during uncertainty of tracked locations. The proposed technique achieves outperforming accuracy with a novel feature selection method and adaptive learning rate for correlation filter based tracker with a conditional switching to the median flow tracker. Later, the work extends combined complementary (discriminative and generative) techniques to track an object in thermal infrared imagery. Finally, the proposed techniques are tested on publicly available benchmark datasets for comparative evaluation.

The thesis also presents a novel vehicle counting algorithm using an object detector combined with the correlation filter based multi object tracker. Results of the proposed algorithm are validated against the manual count.

Contents

List of Figures

List of Tables

NOMENCLATURE

SYMBOL	MEANING
λ	Scalar regularization parameter
$\star$	Convolution
σ	Standard deviation
Ω	Image domain
$I(x, y)$	2D discrete image
$I(u, v)$	2D image in frequency domain
ϵ	Small positive value
$\|.\|$	Number of pixels in a region
$\|\|.\|\|$	Euclidean norm
$L1$	$L1$ norm
min	Minimize the functional
∇	Gradient
μ	Mean
σ^2	Variance
e^x	Exponential function
$\|\Omega\|$	Convolution area
G_σ	Gaussian with variance σ^2
tan^{-1}	Inverse tan trigonometric function
max	Maximize the functional value, maximum
$\odot$	Elementwise multiplication
κ	Kernel function
$\hat{x}$	Discrete Fourier Transform of x
$\Im$	Fast Fouier Transform, Discrete Fourier Transform
$\Im^-1$	Inverse Fast Fouier Transform, Inverse Discrete Fourier Transform
η	Learning rate
$\cap$	Intersection
$\cup$	Union

ABBREVIATIONS

AUC	Area Under Curve
BB	Bounding Box
CBWH	Corrected Background Weighted Histogram
CF	Correlation Filter
CLE	Center Location Error
CN	Color Name
CNN	Convolutional Neural Network
CR	Channel Representations
CSK	Circulant Structure Kernel
D	Dimension
DAT	Distractor Aware Tracker
DC	Direct Current
DF	Distribution Field
DFT	Distribution Field Tracker
DoG	Difference of Gaussian
DP	Distance Precision
DWT	Discrete Wavelet Transform
EDFT	Enhanced distribution Field Tracking
FB	Forward Backword
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
GT	Ground Truth
HoG	Histogram of Gradients
IDCT	Incremental Discriminative Color Tracker
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
KCF	Kernelized Correlation Filter
LBP	Local Binary Pattern
LK	Lucas Kanade
LTIR	Linkoping Thermal InfraRed
MDNet	Multi-Domain Convolutional Neural Network Tracker

MFT	Median Flow Tracker
MIL	Multiple Instance Learning
MOSSE	Minimum Output Sum of Squared Error
NCC	Normalized Cross Correlation
OP	Overlap Precision
OPE	One Pass Evaluation
OTB	Object Tracking Benchmark
OTCBVS	Object Tracking and Classification Beyond Visible Spectrum
P	Precision
PETS	Performance Evaluation of Tracking and Surveillance
PSR	Peak to Sidelobe Ratio
R	Recall
RGB	Red Green Blue
ROI	Region of Interest
RM	Retina Model
SAMF	Scale Adaptive Multiple Feature
SRDCF	Spatially Regularized Discriminative Correlation Filter Tracker
SSD	Sum of Squared Difference
SSQ	Single scale Self Quotient
SSR	Single Scale Retinex
SVM	Support Vector Machine
TLD	Tracking, Learning and Detection
TT	Tan Trigs
VIVID	Video Verification of Identity
VOT	Visual Object Tracking
WF	Weber Face
WLD	Weber Local Descriptor
WN	Wavelet Normalization

Chapter 1

INTRODUCTION

1.1 Introduction

Modern surveillance systems require automated visual analysis such as object and people recognition, action detection, human computer interaction, traffic monitoring, vehicle navigation, etc. For these applications, visual object tracking is a popular problem in computer vision that has been studied by a large community. The primary goal of visual tracking is to estimate the trajectory of an object throughout the frames of a video provided its location in the first frame. Several problems arise due to projection of 3D objects to 2D image space. The common challenges include illumination changes, background clutter, blur, camera motion, scale changes, partial/full occlusion, rotation, deformation, etc. In general, main components of a tracker comprises of feature extraction, target representation, localization, and track management (Maggio *et al.*, 2011). Normally, image regions are expressed using features to describe the appearances of an object. The target region is then localized by optimizing the cost function. The motion model is required to predict the possible locations based on current position of the object in every frame. Further, located target region is utilized to update the appearance model to align with the recent object appearances. The performance of overall algorithm can be enhanced by using improved object representations, motion models, and update techniques. Hence, in this thesis, feature selection, object representation, update techniques, and scale estimation are discussed for color

and infrared videos in tracking perspective. Also, single object tracking is extended to multi target tracking to count the vehicles in highway traffic video. Thus, section 1.2 presents challenges of object tracking. Section 1.3 discusses the motivation and objectives of thesis work followed by the organization in section 1.4.

1.2 Challenges

The detection and tracking are major research areas in computer vision which has been and presently under an exhaustive study. Both detection and tracking are connected in various applications, mainly to surveillance. In addition, the improved quality and resolution of videos have opened diverse applications at lower price. The applications of video tracking include: video analysis for security, surveillance, robotics, human-computer interaction, medical, transportation systems, customer behavior and so on.

Although object tracking is a simple task for humans, it is complicated to realize automated systems. Thus, for an automatic real-time system, the tracking algorithm has to be robust to many challenges such as scale, pose, rotations, moving camera, illumination, occlusion, background clutter, etc. Further, an algorithm capable of keeping track of trajectory record for long-term in real-time would find applications in surveillance and security.

Typically, an object has freedom to move in 3D world which causes *appearance changes* when the object is projected to 2D image space. Such movements include rotation, deformation, and scale change. Non-rigid objects such as animals or humans change appearance due to movement of individual parts. The object also changes its appearance due to relative motion between camera and object resulting in *scale change*. In this situation, tracker needs to keep its location updated to the center of target. In general, an object may move inside or outside environment. Thus, the variation of *illumination* is a common problem due to shadows, lighting, etc. It leads to tracking drift as a result of change in appearance.

Few conditions such as ball flick lead to *abrupt motion* due to which the tracker loses track if the search area is limited to small surroundings. Although there is freedom for an object to move in the 3D space, its 2D projection causes *occlusion* problem

when object moves behind the other object. If the tracker learns from occluded samples, it forgets the actual target that may lead to tracking failure.

The quality of a video is also an important issue for a tracker to be successful. For example, *low resolution* videos typically deficit of texture features, which is the main component required for tracking. Also, focusing error and relative movement of camera or object cause *blurred* video frames. In these cases, tracker fails to capture the spatial features, which is essential for tracking. In this thesis, the major challenges are addressed such as sudden illumination change, occlusion, fast motion, scale change, and deformation.

1.3 Motivation and Research objectives

Object tracking in color and infrared videos are hot research topics that have been and presently under development. Recent benchmark indicates that no object tracking algorithm can handle all the challenges at a time. A popular optical flow based tracker such as median flow tracker Kalal *et al.* (2010) is being used in real time applications. The tracker has provided significant improvement in tracking accuracy. However, the tracker often fails to handle sudden illumination changes in and between the frames. The thesis focuses to combat the illumination related issues by integrating various normalization techniques.

In the recent tracking benchmark, kernelized correlation filter Henriques *et al.* (2015) tracker has shown significant improvement in terms of speed and accuracy. The tracker relies upon texture features with fixed learning rate. In addition, tracker fails to handle texture-less object with deformation, sudden motion etc. These are the motivations for study which suggest feature selection such as color and/or texture, switching to complimentary tracker and adaptive learning rate to overcome tracking drift.

Detection and tracking of objects in thermal imagery have been of interest in several applications mainly for military purposes. The videos acquired by thermal cameras are illumination independent, however they lack texture features which are essential for correlation filter based tracking. Hence, the present work uses efficient

features with complementary approaches to overcome drifting. Automatic counting of vehicles is important nowadays due to inefficient handling of data by humans. Also, these algorithms need to maintain the count of individual vehicles, which help to get the information of traffic density in particular areas. Majority of studies employed fixed camera and simple techniques to achieve the goal. Present work combines You Only Look Once (YOLO) Redmon *et al.* (2016), recent famous object detector with the tracker to serve the purpose. Based on the observation, the following objectives are framed in the thesis:

- To explore the illumination invariant techniques to aid the median flow tracker to achieve drift free tracking during sudden illumination changes.
- To improve the kernelized correlation filter based tracker by incorporating useful features and switching techniques to achieve drift free tracking during occlusion and fast motion.
- To develop an algorithm to track an object in infrared imagery using discriminative and generative approaches.
- To count the vehicles in highway video for traffic management system by combining an object detector and multi target tracking with correlation filters.

1.4 Organization of thesis

In this thesis, improvements of video tracking algorithms in RGB and infrared imagery are addressed. Additionally, an application of tracking is presented for vehicle counting in highway traffic videos. The outline of each chapter is prepared as follows.

Chapter 2 highlights the literature survey on standard algorithms in the field of object tracking in RGB and infrared imagery. Hence, benchmark tracking methods using generative, discriminative, and correlation filter based trackers are presented in this chapter. Also, it provides a background work on vehicle counting application. Further, RGB and thermal infrared benchmark video datasets are reviewed followed by evaluation metrics employed for comparison purpose in the present work.

Chapter 3 presents the detail of median flow tracker, its drawback, and suggests improvements to manage illumination related problems. The focus of this chapter is to provide the state-of-the-art median flow tracker and its improved version especially to boost the performance for rapidly changing illumination videos. Therefore, various illumination normalization techniques, baseline tracker (Kalal *et al.*, 2010), and modified median flow tracker are explained in detail. Experiments are conducted on challenging videos to demonstrate the improved accuracy when compared with the baseline algorithms.

Chapter 4 provides the correlation filter based tracker and suggests an improvement using conditional switching technique. Thus, state-of-the-art kernelized correlation filter (Henriques *et al.*, 2015) based tracker is presented in this chapter. Typically, correlation filter based tracker is more sensitive to deformation, occlusion and fast motion as they learn from spatial features. Hence, a novel feature selection process and an adaptive learning rate are discussed to improve the baseline tracker. Besides, a conditional switching technique is addressed to overcome from tracking drift. Extensive experiments are conducted to analyze the proposed method and compared with the recent trackers.

Chapter 5 exploits the advantages of discriminative and generative approaches to follow an object in infrared imagery. A weighted combination of gradient and channel coded features are considered to locate the object in a kernelized correlation filter based tracker framework (Henriques *et al.*, 2015). In addition, pixel intensities are utilized to localize the object in every frame using AdaBoost classifier. Finally, target location is selected based on position refinement using a generative model. Several experimental results are presented using infrared videos and compared with the existing algorithms.

Tracking finds several applications in traffic management system such as vehicle counting, congestion detection, vehicle speed monitoring, etc. Hence, use of video tracking for measuring the density of vehicles is presented in chapter 6. Thus, vehicle counting is achieved for highway video by combining an object detector with an efficient correlation filter based tracker. Several hand captured highway videos are utilized for testing the proposed method and results are provided to show its accuracy against manual count.

Finally, chapter 7 concludes with the contributions, advantages, and drawbacks of

proposed techniques. Also, possible future directions are indicated.

Chapter 2

VIDEO TRACKING APPROACHES

In the past few years, substantial work has been undergone in the field of visual tracking. The proposed study targets to address the weakness of existing algorithms and aims to improve during conditions such as illumination variation, occlusion, and fast motion. Therefore, this chapter presents theoretical background on video tracking approaches. In literature, mainly two techniques have been discussed namely generative and discriminative. Section 2.1 discusses briefly about standard tracking algorithms using generative techniques. The literature on popular discriminative type trackers is described concisely in section 2.2. Section 2.3 provides background work on correlation filter based tracker. A brief review of object tracking in infrared imagery is presented in section 2.4. Further, state-of-the-art video counting methods are presented briefly in section 2.5. The datasets used for experimentation are described in section 2.6. Finally, quantitative metrics adopted in the work are discussed in section 2.7.

2.1 Generative approaches

Generative type of trackers utilize the appearances of an object to build the target model. The motion of an object is estimated by searching in the confined space for the best match between local patch and model. These techniques are not aimed to

serve long-term tracking, however, maintain some features which assist to search the target.

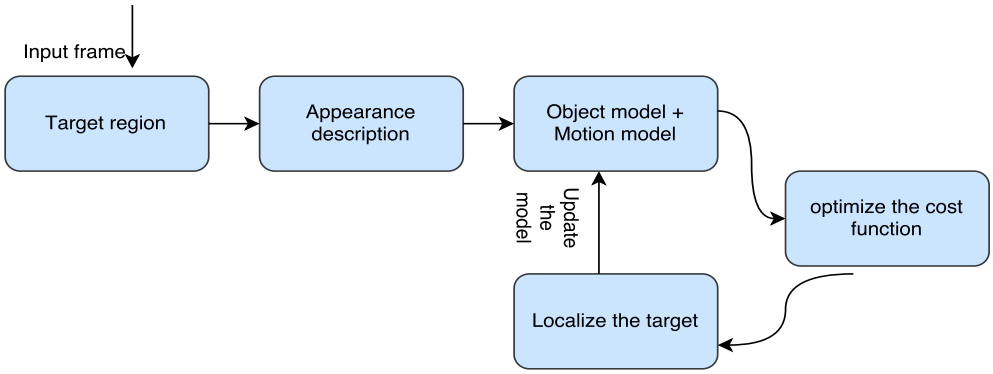


Figure 2.1: Reference model of generative type trackers

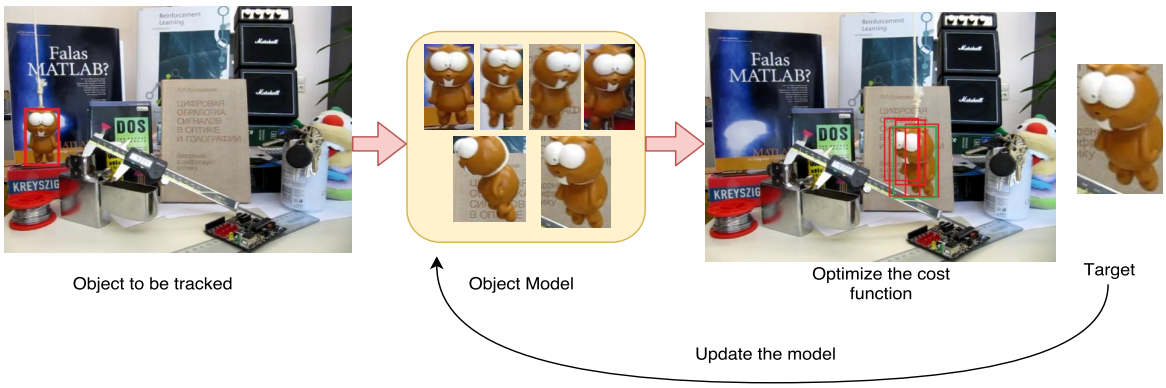


Figure 2.2: Illustration model of generative type trackers

Fig. 2.1 and Fig. 2.2 depict the reference model and illustration of generative trackers respectively. The appearance model is constructed using the features of object samples such as raw pixels (template) (Briechle and Hanebeck, 2001), histogram (Comaniciu *et al.*, 2000), eigen images (Ross *et al.*, 2008), distribution fields (Sevilla-Lara and Learned-Miller, 2012), image patch dictionary (Mei and Ling, 2009). Object is searched locally or globally based on motion model such as Kalman filter or particle filter. The target localization is achieved by minimizing the cost function. Generally, cost function measures the similarity between model and target which include normalized cross correlation (NCC), $L1$ distance, Euclidean distance, etc. The object model is updated in every frame to align with the recent object appearances.

Template matching based trackers express the object appearances in terms of a template. The displacement of an object in consecutive frames is estimated based on the similarity between template and region of interest that maximizes the NCC score (Briechle and Hanebeck, 2001). An exhaustive search in a new frame for the best match is always time-consuming and produces false alarms. Hence, the search is limited to the local neighborhood of previous location. This process can increase the speed, decrease the false alarms but lose target due to large motion of an object. Also, NCC fails to locate the object when there is an occlusion, clutter and appearance changes, which limit its application.

The mean shift tracking is the most popular approach used to search an object in the local neighborhood (Comaniciu *et al.*, 2000). It is a distinct method to avoid an exhaustive search for the best match. In this method, an object is represented using color histogram. The back projection image is constructed through probability distribution of each pixel. Mean shift searches the mode of back-projected image to locate the target. Although color histogram is robust to appearance changes, its distribution may change due to illumination variation. Mean shift approach poorly handles occlusion and background clutter, also there is a chance of the local basin of convergence and loss of spatial information. In addition, a single template model cannot handle partial occlusions. Therefore, fragments based tracker has been proposed by (Adam *et al.*, 2006), where the template is decomposed into multiple arbitrary number of patches. The motion of each fragment is estimated to determine the global motion based on voting statistics. This method can track the object even if two objects of same color are occluded however fails in complex occlusion. Moreover, this technique cannot deal with rotational and scale changes.

Baker and Matthews (2004) proposed a gradient-based approach to find the location of an object in the new frame. This method exploits the sum of squared difference between template and the warped target image to iteratively generate the new set of optimized parameters using the steepest gradient descent algorithm. The affine parameters can deal with scale, rotation, and translation. However, high timing complexity limits its application in real-time scenario.

Typically, template matching based trackers suffer due to considerable appearance variation of an object when the model is not updated. Besides, single template cannot handle multiple appearances. Hence, an extended model is essential to capture the

full range of appearances of target in the past. The principal component images of target appearances are computed on incremental basis over time (Ross *et al.*, 2008). The candidate windows are sampled around the present location based on particle filter motion model, which has Gaussian distribution. The confidence score of each sample is the distance of intensity feature set from candidate window to the target's Eigen image subspace. The candidate with minimum distance is chosen as the target. This method has proved robustness to illumination and appearance changes.

Contrary to template matching, trackers based on bag of visual words use key-point based strategy to detect the image patches in every frame. Two codebooks using RGB and Local Binary Pattern (LBP) features of an image patch are constructed to model the object appearances (Yang *et al.*, 2012). Location of an object is established based on highest similarity between candidate patches and codebooks.

Based on observation, template-based techniques are found to be more sensitive to spatial structures, while histogram features do not provide spatial information. Distribution Field Tracker (DFT) exploits the advantage of template and histogram based techniques to capture the histogram while preserving the spatial location of every pixel (Sevilla-Lara and Learned-Miller, 2012). Thus, the combination of template and histogram based descriptors are used to model the object appearances. The target location is identified based on least distance between the object model and image patches. The proposed method works faster, however, handling occlusion and illumination is still challenging. DFT is further extended by incorporating channel representations to withstand lighting variations and appearance changes (Felsberg, 2013). Thus, Enhanced DFT (EDFT) outperforms the baseline in terms of accuracy and speed.

The limitations of generative trackers are as follows: (i) They consider only appearances of an object to build the model, consequently fails to track due to background clutter. (ii) The model update cannot handle occlusion effectively. (iii) They perform poorly during substantial appearance changes.

2.2 Discriminative approaches

Discriminative type of trackers encode the information of both foreground and background through online learning process. Given the object location and size, algorithms generate a set of binary labeled training samples to update the classifier model. A conventional approach is to treat the tracking as binary classification problem to categorize the region of interest as an object or background. The reference model and illustration of discriminative type tracker are displayed in Fig. 2.3 and Fig. 2.4 respectively.

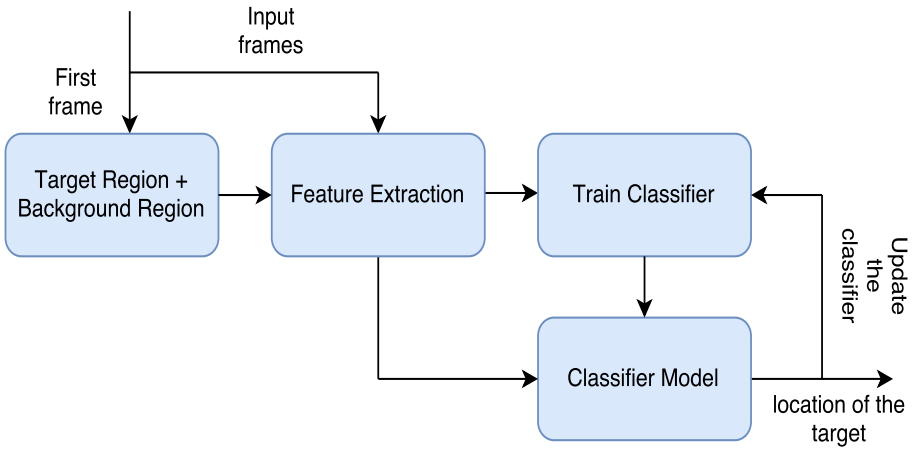


Figure 2.3: Reference model of discriminative type visual tracker

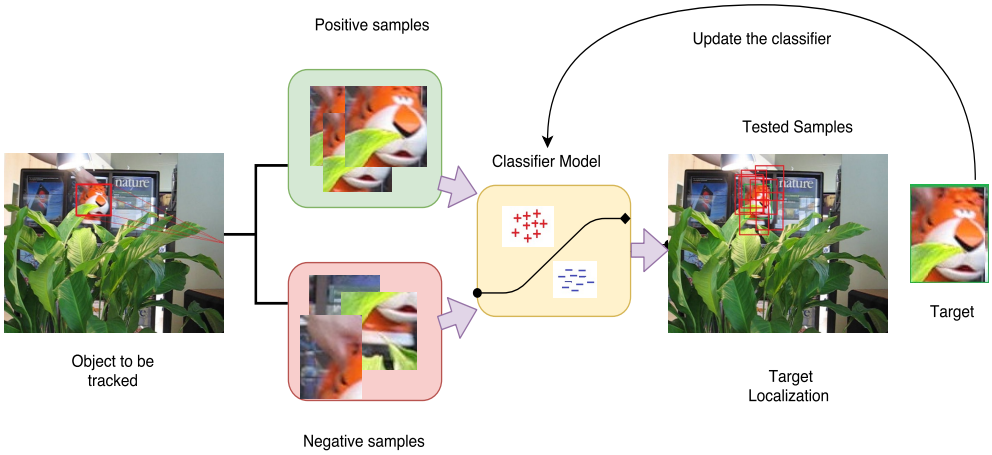


Figure 2.4: Illustration of discriminative type visual tracker

Collins *et al.* (2005) discussed the selection of on-line features for tracking. The system makes use of log-likelihood ratios of class conditional sample densities to distinguish object pixels from background pixels. The mean shift approach evaluates the mode of segmented object region. Although the method is robust to illumination and background clutter, it fails to track during occlusion.

One of the most fundamental discriminative methods integrated off-line Support Vector Machine (SVM) training with optical flow tracking practice (Avidan, 2004). SVM uses horizontal and vertical edge features to distinguish vehicles from its surroundings. The object motion is estimated based on peak confidence score. The shortcoming of this method is that all the appearances of an object have to be captured before tracking begins, which makes it inapplicable to real-time applications. Although the pyramid structure is employed to compensate for significant motion, the method cannot handle partial and complete occlusion of vehicles. Furthermore, there is no assurance that it will shift in case of adjacent vehicles.

To avoid learning from training data in advance, an adaptive discriminative tracker has been introduced. The features of training data are collected on-line by extracting object regions from current tracked location (Avidan, 2007). Thus, each pixel maps to 11D feature vector with eight bin histogram of gradients and R, G, B value. The classifier provides confidence map based on classification score. Mean shift is finally applied to obtain the mode of confidence map to locate the object. However, the tracker is not designed to handle occlusions and large appearance changes.

Similarly, Grabner *et al.* (2006) proposed a technique to design online AdaBoost classifier using Haar-like features, orientation histogram, and local binary pattern as feature set. However, the problem is formulated as a single class classification, hence trained using only positive samples. The target in next frame is located based on maximal confidence score. The suggested algorithm works in real-time and consumes less memory. A co-training model combines generative and discriminative trackers to handle the long-term occlusion (Yu *et al.*, 2008). In this approach, generative model encodes all the appearances while SVM based discriminative tracker uses just recent observations. Although an improvement has been observed in re-detection capability, partial occlusions and large appearance changes were not handled well.

The adaptive discriminative tracker updates the appearances of an object and

background in every frame. The speed of adaptation is crucial in all these systems which decides the amount that it has to forget the old appearances. If the learning rate is too slow or fast, it results in drift. Furthermore, the tracker will gradually ignore the object information due to occlusion. To address this issue, Multiple Instance Learning (MILTrack) (Babenko *et al.*, 2011) proposed a robust method to update the appearance model to manage partial occlusions. It exploits multiple instance learning using training data having a positive bag with at least one positive sample and negative bag with negative samples. The added spatial information reduces the drift significantly. However, if the object is completely occluded for a long period of time or object leaves out of the scene, it starts learning from wrong samples and loses its track. Moreover, articulated objects cannot be handled with this model efficiently.

It has been shown that the object can be tracked precisely by applying the optical flow tracking method (Kalal *et al.*, 2010). Further, the detection unit is integrated to re-initiate the tracking process after occlusion (Kalal *et al.*, 2012). This method combines Lucas Kanade optical flow tracking with random fern based object detection. The system demonstrated impressive performance for long-term videos. Besides, the system was able to determine the extent or to indicate whether the object is present or absent in the video. Even currently, Tracking Learning and Detection (TLD) is being used due to its re-detection capability and real-time speed. However, it does not perform well during out-of-plane rotation and for articulated objects like humans. Later on, kernelized structured support vector machine has been proposed to contribute adaptive tracking (Hare *et al.*, 2016). The combination of Haar features, raw features, and histogram features was used to determine the location of an object in the structured SVM framework. In spite of significant accuracy, the method was not capable of handling the scale and occlusion issues.

An efficient algorithm has been suggested based on features extracted from multi-scale feature space (Zhang *et al.*, 2012). A non-adaptive random projections are used to preserve the structure of image features. A sparse measurement matrix is constructed to compress the sample images comprising foreground and background. The features are used to train a naive Bayes classifier with an on-line update to treat the tracking as a binary classification problem. This method achieved greater accuracy with real-time speed.

Asvadi *et al.* (2013) presented an algorithm based on discriminative 3D RGB his-

togram to classify object and background pixels. The target is located in every frame based on mode of mean shift process. To consider the appearance changes of an object, color learning scheme in adopted. It has been shown that algorithms based on color histograms easily lose target when similar colored object appear nearby. A Distractor Aware Tracker (DAT) makes use of track-by-detection approach based on color appearances (Possegger *et al.*, 2015). A discriminative model is constructed using the color histograms to differentiate object pixels from its surroundings. In addition, DAT suppresses similar regions that appear within the visual field to reduce the tracking drift due to distractor.

Recently, deep learning approaches achieved high accuracy in every field of computer vision. Multi-Domain Convolutional Neural Network (MDNet) tracker (Nam *et al.*, 2016) trains convolutional neural network using a set of videos and ground-truth annotations to construct a generic model for any sequence. This model is applied on new sample around the previous location to identify the target based on maximum classification score. In spite of high accuracy, MDNet requires huge number of off-line videos for training and takes more time for processing.

2.3 Correlation filter based approaches

In contrast, correlation filters have been defined using simple mathematics and achieved high frame rate consuming less memory. The correlation filters are designed to produce peaks for each trained samples in the scene while presenting a low response to the background. The correlation filters are very efficient in detecting and locating the object at faster speed. The widespread applications include object detection, face detection/recognition (Mahalanobis *et al.*, 1987), image registration, object tracking, action recognition (Rodriguez *et al.*, 2008) and so on. The illustration of correlation filter based tracker is depicted in Fig. 2.5.

At first, Minimum Output Sum of Squared Error (MOSSE) filter was introduced to tracking field (Bolme *et al.*, 2010). In this approach, the training data consists of warped object regions to produce Gaussian as desired output as shown in Fig. 2.5. The convolution between filter and area was performed in frequency domain through Fast Fourier Transform (FFT). The peak value of output response determines the

location of target. Thus, the proposed technique was robust to lighting, pose, non-rigid transformation of an object. The increased accuracy and high frame rate of correlation filter tracker attracted research communities to a great extent.

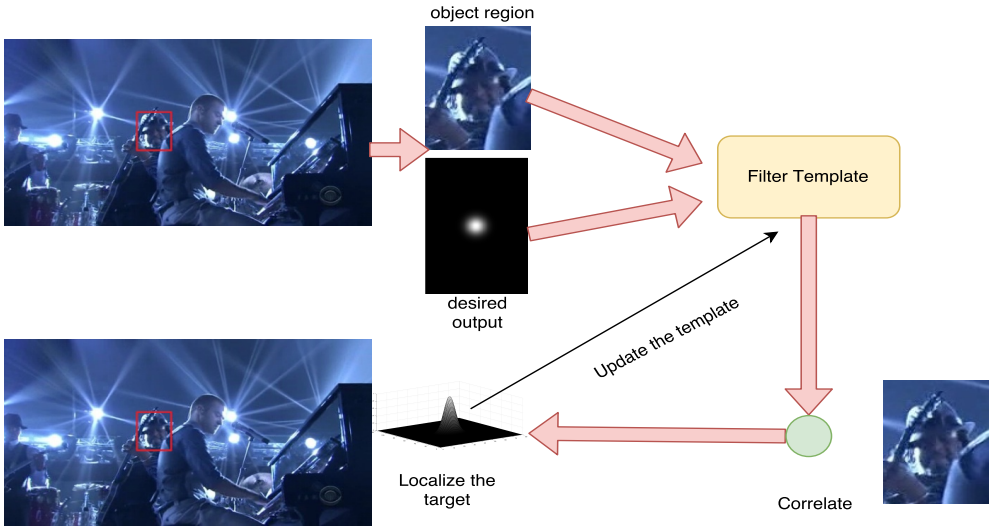


Figure 2.5: Reference model of correlation filter based trackers

Following the success of correlation filters in the field of object tracking, further developments have been suggested. The theory of circulant data structure is exploited to generate a set of positive and negative samples, using which the filter is trained (Henriques *et al.*, 2012). The learning process involved training of data to produce the expected target data as Gaussian template of same size using kernelized classifier. The trained filter template is then convolved with an area cropped from current location in the frequency domain to determine the location of an object. This process is continued in frame-by-frame fashion to accomplish tracking.

Initially, raw pixels were adopted in correlation tracking which is then outstretched to diverse feature space to make it robust. Thus, the histogram of gradient (HoG) feature is utilized in kernelized correlation filter framework for an efficient tracking (Henriques *et al.*, 2015). HoG feature extracts edge details, i.e., texture feature of an object. In contrast, the color name (CN) feature is practiced in the object detection area and is extended for tracking using kernelized correlation filter (Danelljan *et al.*, 2014). However, the above mentioned trackers are not able to handle the scale of an object.

To estimate the size of an object, detection procedure is implemented at multiple scales. Accordingly, an efficient scale adaptive tracking has been proposed. A scale space object template is constructed to find the peak in correlation responses (Li and Zhu, 2014). The position and scale corresponding to maximum value are identified as the target. A separate 1D correlation filter has been proposed to tackle the scale more effectively (Danelljan *et al.*, 2014). The scale filter is trained using variable size templates. The scale corresponding to peak value of 1D correlation provides the required scale. Both translational and scale filters are updated with new samples in every frame. However, the advantages of baseline tracker are lost due to increase in the computational cost.

Correlation filter based trackers usually suffer from the inaccurate training samples due to circular correlation. Spatially Regularized Discriminative Correlation Filter (SRDCF) (Danelljan *et al.*, 2015) introduces spatial regularization to penalize the filter coefficients lying outside the object region. The spatial regularization function is selected to optimize the filter coefficients in the Fourier domain through iterative Gauss-Seidel optimization procedure. A more precise location is estimated through sub-grid approximation. Different versions of SRDCF use grayscale, HoG, CN and covolutional neural network (CNN) features.

2.4 Tracking in infrared imagery

The visible spectrum is the only part of electromagnetic spectrum that individuals can see. All objects in the world has property to reflect, absorb or transmit energy. Normal visual cameras use reflected light to acquire images for which it expects light source. In contrast, thermal infrared (TIR) cameras capture images based on temperature of an object that is radiated, hence do not need any lighting source. Infrared technology was initially employed in the military area and consequently moved to various fields of industry, scientific and medical areas. Therefore, night vision cameras are extensively found in applications such as building inspection, gas detection, industry, medical, veterinary, agricultural, fire detection, surveillance, aerospace, target acquisition, tracking of humans, vehicles, and animals (Gade and Moeslund, 2014). The power of thermal cameras to capture images in all climate conditions and dimness make considerable impact on its applications.

Infrared (IR) is invisible electromagnetic radiation with longer wavelengths than that of visible spectrum. Infrared band is divided into different ranges based on wavelength. The infrared range is divided into various bands that include near infrared (0.7 μm to 1 μm wavelength), short-wave infrared (1 μm to 3 μm wavelength), mid-wave infrared (3 μm to 5 μm wavelength) and long-wave infrared (8 μm to 1.4 μm wavelength). Near infrared or visual cameras capture radiations reflected by objects, while long-wave or mid-wave infrared cameras capture radiations emitted by objects. Tracking of an object in infrared imagery is complicated due to following reasons:

- Infrared images appear noisy with low signal to noise ratio (SNR), poor contrast, and low resolution comprising large number of dead pixels.
- Visual aspect of infrared images is similar to gray-scale images with missing texture and color features.
- The intensity of object varies with temperature instead of illumination in color images.
- Occlusion and re-identification of objects in thermal infrared imagery are challenging task as two objects of comparable size, shape and color look very similar. Examples include individuals strolling in group, animals of same sort, fowls in a rush share similar shape and intensity level in thermal imagery.
- The detection and tracking are more challenging for complex background scenes as the object may blend with surroundings, change size, shape, and intensity.
- Overall, it is difficult to find the unique property of an object in infrared imagery.

Thermal cameras have been originally developed as a night vision tool for surveillance, military, and later extended to a wide variety of applications (Gade and Moeslund, 2014). These passive sensors eliminate the illumination related problems arising in RGB videos. The detection and tracking of objects in infrared imagery find an important role in several applications such as military, surveillance, and so on. In literature, many methods have been proposed in thermal infrared tracking, that include pedestrian or object as target. In the past, tracking algorithms proposed for color videos have been employed to track an object in thermal infrared imagery (Felsberg

et al., 2015). Similar to RGB tracking, generative and discriminative type trackers have been utilized in thermal infrared tracking.

Tracking of pedestrian in infrared videos has been achieved by integrating intensity and edge cues in a particle filter framework using an adaptive integration scheme (Wang and Tang, 2010). Automatic updating strategy further increases the performance accuracy. Wang *et al.* (2012) applied Gaussian Mixture Model (GMM) to build the efficient background model in order to separate the foreground from background. Besides, Support Vector Machine (SVM) classifier is trained using shape features to detect the pedestrians. Following detection, intensity combined with edge cues under particle filter is utilized to track the pedestrians in infrared videos. Lamberti *et al.* (2011) exploited motion prediction techniques to find possible false alarms and activates template matching for recovery purpose. Thus, the activation strategy has high impact on improved accuracy. A combination of curve matching with Kalman filter has been proposed by (Lee *et al.*, 2012) to predict the position of target in infrared video. An improvement of tracking accuracy has been observed due to weighted mean of two methods compared to Kalman filter approach.

The detection of small infrared object has been proposed by (Dong *et al.*, 2014*a*) using R-means clustering technique to cluster the interest points corresponding to foreground and background. Dong *et al.* (2014*b*) applied Difference of Gaussian (DoG) filters for enhancement and proportional integral derivative to predict the location of object. A combination of detection and filtering approach has been employed to track the objects (He *et al.*, 2015) which exploits object detection based on sparse representation and filtering through weighted correlation filter. Terravic Motion IR Database of Object Tracking and Classification Beyond Visible Spectrum (OTCBVS) (`http://vcipl-okstate.org/pbvs/bench/`) and Video Verification of Identity (VIVID) datasets (`http://vision.cse.psu.edu/data/vividEval/datasets/datasets.html`) are utilized to validate the proposed method. Subsequently, ABCD (4 approaches named as A, B, C, and D) tracker extends EDFT (Felsberg, 2013) tracker to select the object region adaptively (Berg *et al.*, 2016). In addition, the background information is exploited to avoid its contamination in the object model. Moreover, it estimates the scale of object based on probability mass change.

2.5 Vehicle counting

A number of algorithms exist to count the vehicles using video data. In this section, we review the state-of-the-art vehicle counting techniques for videos captured using still cameras.

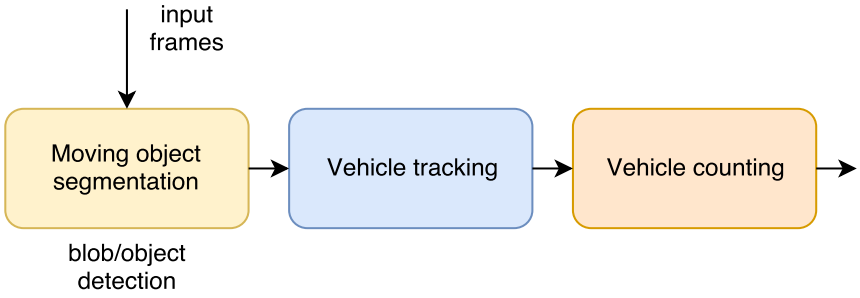


Figure 2.6: Reference model of vehicle counting

The process of vehicle counting involves extraction of moving objects, tracking and counting based on rules as shown in Fig. 2.6. Counting of vehicles in day and night time has been proposed using spatio-temporal analysis and morphological analysis of head lights respectively (Cucchiara *et al.*, 2000). Two level architecture comprising of low level image processing tools to extract the vehicle data and high level module as a forward chaining production rule to suit for urban traffic. The system was flexible to count the vehicles in 24 hour video and produced good accuracy. An approach by (Barcellos *et al.*, 2015) uses particle filter to obtain the particles with motion coherence and spatial adjacency. Later, groups of particles associated with similar motion patterns are exploited to detect the moving vehicles. However, the particles are sampled based on foreground mask generated using GMM and motion energy images which provide the possible vehicle location. Later on, color property is used to measure the similarity of vehicles in adjacent frames. The counting is accomplished in a user-defined loop and shown improved accuracy. However, heavy occlusions may deteriorate the counting accuracy. Also, the system fails to count small vehicles and during traffic flow interruption. Illumination and shading effects create artifacts resulting in false detection. Also, vehicles are not categorized in this technique. To detect the vehicles during occlusion, a windshield based hypothesis is developed (Van Pham *et al.*, 2015). Accordingly, Hough transform is utilized to detect the trapezoidal like structure to

identify the probable location of vehicles. HoG features of well-collected datasets are used to train SVM classifier to verify the vehicles at later stage. Tracking is performed using Kalman filter to estimate the trajectory and simple rule-based reasoning is employed to count the vehicles.

Xia *et al.* (2016) proposed to fuse expectation maximization (EM) algorithm with GMM to segment the moving vehicles in a user-defined virtual loop. A restoration method is used to remove noise and fill gap to extract the vehicle region. The occlusions of vehicles are handled using morphological feature and color histograms. The algorithm has been tested on videos captured using still camera at the road intersections. An active basis model and symmetry property are employed to detect the vehicles in a highway video (Kamkar *et al.*, 2016). Additionally, vehicles are classified by training random forest classifier with time-spatial image and correlation computed from gray level co-occurrence matrix. A highway vehicle counting in compressed domain is accomplished using hierarchical classification based regression (Liu *et al.*, 2016). In this, the system extracts low-level features to train two-layer classifier. The method can categorize the scene into heavy, medium, and light based on traffic density. In all these methods, it is observed that still/stable cameras have been used to acquire the video. Besides, the video contains single lane and simple background. However, vehicle counting process for shaking or hand recorded videos is a tedious task.

2.6 Datasets

Since the last decade, tracking area has significantly grown due to its widespread applications. Five years before, tracking algorithms were suffering from insufficient video data and performance evaluation metrics to assess the advances in field. The several datasets have been released in the past for RGB videos such as Performance Evaluation of Tracking and Surveillance (PETS) (`www.cvg.reading.ac.uk/PETS2015/`), Visual Object Tracking (VOT) challenges (`www.votchallenge.net/`) and Object Tracking benchmark (OTB) (`cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html`) and Linkoping Thermal InfraRed (LTIR) (`www.cvl.isy.liu.se/en/research/datasets/ltir/`), OTCBVS (`http://vcipl-okstate.org/pbvs/bench/`) and VIVID datasets

(`http://vision.cse.psu.edu/data/vividEval/datasets/datasets.html`) for infrared videos. This thesis uses a subset of OTB (Wu *et al.*, 2013) and LTIR datasets (Berg *et al.*, 2015) for experimental analysis of RGB and infrared target tracking respectively. The detail of each video is provided in the following subsections briefly.

Table 2.1: Challenges associated with 17 sequences from OTB dataset

Sequence	**No of Frames**	**Target**	**Challenges**
Deer	71	deer	motion blur, fast motion, rotation, clutter
Bolt	350	person	rotation, clutter, deformation, scale
Boy	602	person	scale, motion blur, fast motion, rotation
Jogging-1	307	person	occlusion, deformation, rotation, scale
Jogging-2	307	person	occlusion, deformation, out-of-plane rotation, scale change
Doll	3872	doll	illumination, deformation, rotation
David3	252	person	occlusion, deformation, rotation, clutter
Jumping	313	person	motion blur, fast motion
Dog1	1350	doll	scale, rotation
Lemming	1336	doll	illumination, scale, occlusion, fast motion, rotation, out-of-view
Basketball	725	person	clutter, rotation, deformation, occlusion, illumination
Subway	175	person	deformation, occlusion, clutter
Tiger1	354	doll	occlusion, deformation, motion blur, fast motion, rotation, illumination
Crossing	120	person	scale, deformation, clutter
Couple	140	persons	scale, deformation, fast motion, rotation
Shaking	365	person	rotation, clutter, scale, illumination
Surfer	376	person	fast motion, rotation, low resolution, scale

2.6.1 OTB datasets

OTB dataset contains a large number of videos collected from various sources. They have been annotated manually using attributes based on common challenges in visual tracking. The challenges include illumination variation, scale changes, deformation, motion blur, in-plane and out-of-plane rotation, out of view, background clutter, low resolution, occlusion, etc. Each dataset is associated with challenging aspects, and ground-truth file with each row representing the bounding box of target in that frame i.e., (x, y, width, height). In this thesis, a subset of OTB dataset is considered to compare algorithm's performance.

2.6.2 LTIR datasets

LTIR dataset gathers videos from different sources captured by several thermal sensors. The sequences given in the dataset cover indoor and outdoor videos recorded under different climate conditions. The dataset includes challenges and ground-truth annotations that provide center coordinates (i.e., (x, y)) of the object bounding box, and its size, i.e., (x, y, width, height).

Table 2.2: Challenges associated with 17 sequences from LTIR dataset

Sequence	**No of Frames**	**Target**	**Challenges**
Birds	270	person	moving camera, occlusion, scale change
Boat1	625	boat	camera motion, scale change
Boat2	951	boat	camera motion, scale change
Crouching	618	person	occlusion
Depthwise crossing	851	person	scale change
Dog	92	dog	camera motion, occlusion, scale change
Garden	676	person	camera motion, occlusion, scale change
Hiding	358	person	camera motion, occlusion, scale change
Jacket	1451	person	occlusion, scale change
Quadrocopter	178	quadrocopter	camera motion, motion blur
Ragged	1035	boat	camera motion, scale change
Running rhino	763	rhino	camera motion
Saturated	218	person	camera motion
Soccer	235	person	motion change
Selma	775	person	scale change
Street	172	person	camera motion, scale change
Trees1	665	person	camera motion, occlusion

2.7 Evaluation metrics

Tracking evaluation metrics are used to assess the tracking performance. Various authors prefer different metrics. In this thesis, metrics used by majority of research groups are considered (Wu *et al.*, 2013). For all experiments, evaluation of tracking algorithms include three metrics: Average Center Location Error (ACLE), Distance Precision (DP) score and Overlap Precision score (OP). Center Location Error (CLE) is defined as Euclidean distance between tracked location and ground-truth location.

Let (x_t, y_t) denotes tracked location, (xg_t, yg_t) represents ground-truth location at frame t, N is the total number of frames in a video, then average center location error is obtained as

$$ACLE = \frac{1}{N}\sum_{t=1}^{N}\sqrt{(x_t - xg_t)^2 + (y_t - yg_t)^2}. \tag{2.1}$$

This metric is very popular and works well for point targets. However, the measure depends on object size, and not consistent with the object shape. It becomes difficult to define the center for articulated objects like humans. In addition, CLE becomes significantly large when the tracker drifts off. Thus, to measure the tracking performance, successful frames are considered based on CLE. DP is computed to determine the percentage of number of frames with center location error less than threshold (i.e. $T_{DP} = 20$ pixels). Thus, DP is obtained as,

$$DP = \frac{1}{N}\sum_{t=1}^{N}\delta\left(DP_t \leq T_{DP}\right), \tag{2.2}$$

where, DP_t denotes Euclidean distance between tracked location and ground-truth location at frame number t.

OP is another metric to evaluate the tracker, which is widely used among researchers of tracking community. OP uses bounding box overlap, which is defined as the overlap between tracked region and ground-truth region as, $S_t = \frac{|BB_t \cap GT_t|}{|BB_t \cup GT_t|}$, where BB_t represents tracked bounding box, GT_t denotes ground-truth bounding box, $\cap$ represents the intersection, $\cup$ denotes the union of two bounding boxes and $|.|$ denotes the number of pixels in the area bounded by a region. To measure the overlap precision score of a tracker, percentage of number of successful frames whose overlap S_t greater than certain threshold (i.e. $T_{OP} = 0.5$) is computed as

$$OP = \frac{1}{N}\sum_{t=1}^{N}\delta\left(S_t \geq T_{OP}\right), \tag{2.3}$$

where, S_t denotes overlap score of frame t.

OTB benchmark quantifies tracking results graphically in the form of precision plot and success plots. Precision plot is based on CLE, that describes the plot of an average number of frames having distance precision score within the threshold values chosen

in the range of 0 to 50 pixels. Similarly, success plot is depicts an average number of successful frames with bounding box overlap greater than thresholds selected in the range of 0 to 1. Finally, Area Under Curve (AUC) is utilized in order to rank the algorithms.

All experiments have been conducted using initial ground-truth location for one round. Hence, One Pass Evaluation (OPE) method is adopted to compare the proposed method with other trackers.

2.8 Summary

In this chapter, we have presented state-of-the-art tracking methods developed in the field of visual tracking in RGB videos and thermal infrared videos. Thus, the generative and discriminative tracking approaches have been discussed. It has been observed that, there is no single method which provides good accuracy for given set of videos. Each algorithm is associated with certain advantages and shortcomings. Thus, it fails to track for a particular video sequence. Similar to RGB videos, thermal videos find numerous applications in military, surveillance, medical, and industry area. Hence, many standard algorithms proposed in the field have been briefly reviewed. Video tracking plays important role in people or vehicle counting applications. Recently, video based traffic management system has become more popular. Consequently, well-known vehicle counting methods have been reviewed.

The popular benchmark datasets for object tracking in RGB and infrared videos have been addressed. In addition, details of image sequences utilized in this thesis have been presented. The quantitative metrics used to assess the video tracking approaches have been discussed at the end.

Chapter 3

ILLUMINATION INVARIANT OBJECT TRACKING

In this chapter, two approaches are presented to attain illumination invariant tracking. Although the state-of-the-art tracking technology is rapidly growing, few issues are still hard such as illumination variation, occlusion, deformation, out-of-view, motion blur, etc. Among these challenges, sudden illumination variation is more complicated which is not handled effectively by several trackers. Majority of them, indeed work under controlled illumination conditions in indoor and outdoor context. However, sudden illumination variations affect the performance of tracker. This chapter suggests, binding a photometric normalization technique with illumination sensitive tracker to reduce the drift due to unexpected light variation. The tracker under study is median flow tracker (MFT) (Kalal *et al.*, 2010) which applied optical flow method to track an object and produced remarkable results in the tracking history. However, the tracker drifts off due to sudden light variation. To solve this difficulty, pre-processing technique is incorporated just before tracking. Hence, in this chapter, two methods are explained. They cover (i) extracting the reflectance component (illumination independent) of a video frame and utilize it for tracking (ii) maintain uniform illumination throughout the video irrespective of light variations. In the first approach, illumination invariant component is employed for tracking an object. Thus, section 3.1 describes the need for photometric correction techniques in video tracking algorithms. Median flow tracker and its modified versions with experimental details are presented

in section 3.2.

In the second approach, illumination of a video is kept constant by altering discrete cosine transform (DCT) coefficients of an image in the logarithmic domain. The slowly varying illumination component is mainly indicated by low-frequency coefficients of an image. Accordingly, a fixed number of DCT coefficients is ignored. Furthermore, DC coefficient is maintained almost fixed all through the video to minimize the effects of sudden shift in brightness values. Besides, each video frame is enhanced by employing pixel transformation technique that improves the contrast of dull images based on probability distribution of pixels. Thus, the work focuses on handling the gradual and abrupt changes in the illumination. Section 3.3 explains the procedure to maintain constant brightness of a video with experimental analysis. Finally, section 3.4 summarizes illumination invariant median flow tracker.

3.1 Illumination normalization techniques

Illumination variation is a common concern in real-time situation, where light intensity tends to change due to shadows in an outside environment and flashlights in an indoor setting. There are two main research paths aimed to tackle illumination related issues. They involve either appending a pre-processing stage to illumination sensitive trackers or extracting illumination invariant features needed for tracking. The tracking algorithms that employ direct pixel values, histogram features, optical flow, sub-space techniques and dictionary-based approach mainly suffer from drifting due to sudden illumination changes. Several successful state-of-the-art trackers such as MFT (Kalal *et al.*, 2010), L1 tracker (Mei and Ling, 2009), CT (Zhang *et al.*, 2014), DFT (Sevilla-Lara and Learned-Miller, 2012), TLD (Kalal *et al.*, 2012) are enabled to handle various challenges in video tracking. However, these methods are not robust to abrupt changes of light. Hence, in this chapter, we consider popular median flow tracker (Kalal *et al.*, 2010) for all variety of illumination changes and incorporate illumination normalization techniques to improve the tracking result.

The photometric normalization techniques are being used in the face recognition systems to offset the appearance variation due to light conditions (Ochoa-Villegas

et al., 2015),(Han *et al.*, 2013). In object tracking context, illumination variation creates an enormous variation of appearances. Numerous ways have been introduced in the past to compensate the illumination consequences that include simple image enhancement to denoising techniques. Modified $L1$ tracker (Nhat *et al.*, 2014) employs a wavelet-based normalization technique to compensate for light variation, Phadke *et al.* (2017) advise to use modified LBP feature in mean shift tracking to manage illumination challenges. The popularity of these pre-processing methods lies in its simplicity, processing speed and improved accuracy which helps to incorporate before tracking more effectively. Numerous works on photometric normalization methods have been reviewed in the face verification (Han *et al.*, 2013), (Ochoa-Villegas *et al.*, 2015), and eye gaze tracking (Armato *et al.*, 2013). However, there is need of knowledge to apply these techniques for object tracking. Therefore, this chapter presents a comparison of most efficient photometric normalization techniques when used as pre-processing stage for solving illumination related problems using median flow tracker.

This section describes the principle of image acquisition. The retinex theory (Land *et al.*, 1971) explains about human visual system, which can distinguish colors from the reflectance of a scene by discarding illumination. Image of a scene can be represented as $I(x, y) = R(x, y)L(x, y)$, where $I(x, y)$ denotes an image, $R(x, y)$ indicates the reflectance and $L(x, y)$ expresses the illumination component of a scene. Reflectance component characterizes the details of an object, while illumination source decides the luminance component. In video tracking context, object movement needs to be reported in continuous frames irrespective of illumination changes. Let $I_t(x, y)$ and $I_{t+1}(x, y)$ denote video frames at t and $t + 1$ respectively. The reflectance component is stationary for given scene, hence according to retinex theory,

$$I_t(x, y) = R_t(x, y)L_t(x, y). \tag{3.1}$$

Applying logarithmic transform gives,

$$\ln \mathrm{I}_t(x, y) = \ln \mathrm{R}_t(x, y) + \ln L_t(x, y) \tag{3.2}$$

and

$$\ln \mathrm{I}_{t+1}(x, y) = \ln \mathrm{R}_{t+1}(x, y) + \ln L_{t+1}(x, y). \tag{3.3}$$

The reflectance of scene remains constant, i.e., $\ln \mathrm{R}_t(x, y) = \ln \mathrm{R}_{t+1}(x, y)$. Therefore,

Eq. 3.3 becomes

$$\ln \mathrm{I}_{t+1}(x,y) = \ln \mathrm{R}_t(x,y) + \ln L_{t+1}(x,y). \tag{3.4}$$

If the illumination component at frame t and $t+1$ differs by ϵ then Eq. 3.4 can be written as,

$$\ln \mathrm{I}_{t+1}(x,y) = \ln \mathrm{R}_t(x,y) + \ln L_t(x,y) \pm \epsilon(x,y), \tag{3.5}$$

hence,

$$\ln \mathrm{I}_{t+1}(x,y) = \ln \mathrm{I}_t(x,y) \pm \epsilon(x,y). \tag{3.6}$$

The next frame I_{t+1} contains pixel values with different illumination. The normalized image is acquired from the original frame by compensating the additive term $\epsilon(x,y)$ (Phadke *et al.*, 2013). In this context, we extract the reflectance component of an image which is independent of illumination and it is derived based on some assumptions. They include, illumination corresponds to slowly varying component and reflectance corresponds to sharp edges of an image. There are numerous techniques available to smooth the image that includes Gaussian filtering, non-local means filtering, total variation model filters, isotropic, and anisotropic filtering. These methods determine the illumination component, which is then subtracted from the image in logarithmic domain to obtain the reflectance component.

Illumination normalization algorithms used in the proposed study are given in Table 3.1 and briefly explained in the following subsections. To understand the effect of photometric normalization techniques, we apply on two separate video frames 58^{th} and 59^{th} of *shaking* video from OTB dataset with varying illumination component between adjacent frames. Original images are converted to gray scale and displayed in Fig. 3.1.

Figure 3.1: *shaking* sequence frames 58 (low illumination) and 59 (large illumination) which exhibits the flashlight effects in an indoor setting

Table 3.1: List of the photometric normalization methods employed in the proposed tracker.

Research Paper	**Normalization Method Used**	Time(s)
Single Scale Retinex (SSR) (Jobson *et al.*, 1997)	Subtracts blurred version (using the Gaussian filter) of an image from the original image in logarithmic domain	0.75
Single scale Self-Quotient Image(SQI) (Wang *et al.*, 2004)	Divides the image by blurred version of an image	8.29
Wavelet based Normalization (WN) (Du *et al.*, 2005)	Divides the image into low and high frequency sub-bands using DWT in logarithmic domain. Histogram equalization is applied to low frequency component and high frequency components are amplified to enhance the edges	1.86
Discrete Cosine Transform (DCT) (Chen *et al.*, 2006)	The logarithm of an image is converted to frequency domain using DCT. Low frequency coefficients are discarded to nullify the illumination effects	1.08
Tan and Triggs (TT) (Tan and Triggs, 2010)	Uses a series of operations such as gamma correction, difference of Gaussian filtering, optional masking and contrast equalization	0.29
Weber Face (WF) (Wang *et al.*, 2011)	Computes relative gradient, which is the ratio of difference between current pixel and neighbors to the current pixel	1.21
Retina Model (RM) (Vu and Caplier, 2009)	Mimics human retina model to implement 3 layers i.e, photo receptors, outer plexiform layer and inner plexiform layer. It uses Gaussian low pass filters and DoG filter to normalize the image	0.11
Difference of Gaussian (DoG) (Struc, 2012)	A band pass filter is applied to extract the reflectance component of an image	0.1

3.1.1 Single Scale Retinex (SSR)

Jobson *et al.* (1997) proposed a method to separate luminance and reflectance component from an image based on retinex theory. In this method, reflectance component is estimated by subracting the blurred version of an image from original image in the

logarithmic domain. Therefore, the reflectance component is obtained as

$$\log R(x,y) = \log I(x,y) - \log(S(x,y) \star I(x,y)), \tag{3.7}$$

where, $\star$ is a convolution operator, $S(x,y)$ is the center surround system which take Gaussian function for smoothing as given by

$$S(x,y) = Ke^{-\frac{\sqrt{x^2+y^2}}{c^2}}, \tag{3.8}$$

where, c is the bandwidth of Gaussian, K is a constant, and is computed such that $\iint S(x,y)dxdy = 1$. In Eq. (3.7), $R(x,y)$ stands for reflectance component of an image. Fig. 3.2(a) shows *shaking* video frames 58 and 59 after applying SSR algorithm. The scale of Gaussian filter is set at 15.

3.1.2 Single scale Self Quotient Image (SSQ)

According to Lambertian model, the image factors into intrinsic and extrinsic components. Self-quotient image (Wang *et al.*, 2004) is the intrinsic factor, derived based on assumptions like human vision is sensitive to reflectance and insensitive to illumination. Moreover, human vision reacts to local contrast than global illumination. Similar to retinex method, self-quotient image is applied for photometric normalization. An illumination invariant image is obtained as follows:

$$Q(x,y) = \frac{I(x,y)}{I_s(x,y)} \tag{3.9}$$

and

$$Q(x,y) = \frac{I(x,y)}{F \star I(x,y)}, \tag{3.10}$$

where, $I_s(x,y)$ is the smoothed version of I and F is the smoothing mask similar to SSR. Compared to SSR, structure of smoothing kernel F is modified using the weight function. Thus, for every convolution region, a filter is formed as $F = GW$, where W denotes weight, and G represents Gaussian kernel. A convolution region Ω is separated into two sub-regions M_1 and M_2 based on threshold value τ, which is calculated as the mean of pixel values of convolution region. The weight function is obtained for

Gaussian smoothing filter as

$$W(x,y) = \begin{cases} 1 \text{ if } I(x,y) \in M_1 \\ 0 \text{ if } I(x,y) \in M_2, \end{cases} \tag{3.11}$$

such that

$$\frac{1}{M} \sum_{\Omega} W(x,y) G(x,y) = 1, \tag{3.12}$$

where M is normalizing factor. The edge region has a large gray value variation in convolution region and threshold divides convolution region into 2 sections M_1 and M_2 as given by

$$I(x,y) \in \begin{cases} M_1 \text{ if } I(x,y) \leq \tau \\ M_2 \text{ if } I(x,y) > \tau \end{cases} . \tag{3.13}$$

The filter smooths main part of convolution region while preserving discontinuities. The bandwidth of Gaussian filter is fixed at 1. SSQ based illumination normalized images are depicted in Fig. 3.2(b) utilizing *shaking* video frames 58 and 59. It shows that SSQ technique eliminates illumination effects by highlighting edge features.

3.1.3 Wavelet based illumination normalization techniques (WN)

Du *et al.* (2005) proposed a method to improve the contrast of image in the frequency domain using wavelet transform. In this technique, an image is decomposed into low-frequency and high-frequency components using 2D wavelet filter. Thus, the image is split into four sub-bands such as LL, LH, HL, and HH. Wavelet transform can be further applied in depth to obtain multi-stage components. Subsequently, histogram equalization is applied to low-frequency coefficients while high-frequency components are magnified by multiplying with a scalar greater than 1. Finally, normalized image is acquired via re-construction using inverse discrete wavelet transform (IDWT). Results of wavelet-based normalization technique are displayed in Fig. 3.2(c) for *shaking* video frames 58 and 59.

3.1.4 DCT based normalization (DCT)

Chen *et al.* (2006) suggested a method to obtain the reflectance by altering DCT coefficients. The technique assumes that the illumination changes are gradual as compared to reflectance part. Accordingly, low-frequency coefficients of an image are discarded. Initially, logarithm of an image is transformed to frequency domain using DCT. DC coefficient is replaced by the mean value, and few low-frequency coefficients are discarded by preserving high-frequency components. The DC coefficient is set to

$$C(0,0) = \log(\mu)\sqrt{MN}, \tag{3.14}$$

where M and N denote the size of image, μ represents the mean value of image. 25 low-frequency coefficients (top left corner values) are changed to zero by arranging them in zig-zag order. Finally, inverse discrete cosine transform (IDCT) is applied to retrieve reflectance image. The frames 58 and 59 of *shaking* video using DCT technique are shown in Fig. 3.2(d). It is observed that DCT technique nullifies illumination impacts while preserving the main details of image.

3.1.5 Retina Model based normalization (RM)

Retina model mimics the human retina by combining two adaptive non linear functions and difference of Gaussian filter for normalizing images in face recognition system (Vu and Caplier, 2009). The first non linear function is computed as

$$f_1(p) = I_i(p) * G_{\sigma 1} + \frac{\overline{I_i}}{2}, \tag{3.15}$$

where p represents each pixel, $f_1(p)$ is the adaptation factor for pixel p, I_i denotes input image, $\star$ represents convolution operator, $\overline{I_i}$ denotes the average value of image and $G_{\sigma 1}$ is 2D Gaussian filter given by $G_{\sigma 1}(x,y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}}$, where σ_1 is the standard deviation. The input image is then processed using the adaptation factor as

$$I_1(p) = (I_{i\max} + f_1(p))\frac{I_i(p)}{I_i(p) + f_1(p)}, \tag{3.16}$$

where I_{imax} denotes maximum value of the image intensity. The second non linear function is obtained by

$$f_2(p) = I_1(p) * G_{\sigma 2} + \frac{\overline{I_1}}{2} \tag{3.17}$$

and $G_{\sigma 2}(x, y) = \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}$ where σ_2 denotes standard deviation of Gaussian function $G_{\sigma 2}$. $\sigma_1 = 1$ and $\sigma_2 = 3$ are selected for Gaussian functions $G_{\sigma 1}$ and $G_{\sigma 2}$ respectively. The second nonlinear function is applied to get the image I_2 as

$$I_2(p) = (I_{1\max} + f_2(p)) \frac{I_1(p)}{I_1(p) + f_2(p)}. \tag{3.18}$$

It is then convolved with Difference of Gaussian filter (DoG) to obtain $I_b = DoG * I_2$, where $DoG = G_{\sigma L} - G_{\sigma H}$, and σL =0.5, σH =4 respectively denote the standard deviation of Gaussian low pass filters corresponding to photoreceptors and horizontal cells. The normalized image is then obtained as $I_n(p) = \frac{I_b(p) - \mu_{I_b}}{\sigma_{I_b}}$. The final image is then enhanced by truncating based on threshold as

$$I_f = \begin{cases} \max(T, |I_n(p)|) \text{ if } I_n(p) \geq 0 \\ -\max(T, |I_n(p)|) \text{ else} \end{cases}. \tag{3.19}$$

The threshold T is set at 5. The results of applying retina model on *shaking* video frames are depicted in Fig. 3.2(e) and shows its ability to remove illumination component while keeping the texture of an image unchanged.

3.1.6 Tan Triggs normalization technique (TT)

To combat the illumination effects and shadow in an image, a chain of pre-processing techniques has been (Tan and Triggs, 2010) proposed. A nonlinear (gamma γ) transformation I^γ of each pixel in an image I has been used to enhance the local dynamic range in dark areas and compress the dynamic range in bright areas. Since gamma correction does not remove shading effects, band pass filter using Difference of Gaussians (DoG) is applied to retain fine details and remove low-frequency illumination component. The contrast of image is then improved by rescaling the image intensities

in two-stage process given by

$$I(x,y) = \frac{I(x,y)}{(mean\left(|I(x',y')|^{\alpha}\right))^{\frac{1}{\alpha}}} \tag{3.20}$$

and it can be written as,

$$I(x,y) = \frac{I(x,y)}{(mean\left(\min\left(\beta, |I(x',y')|^{\alpha}\right)\right))^{\frac{1}{\alpha}}}. \tag{3.21}$$

In the above equation, α reduces the effect of large values and β is used to remove large values after normalization. Finally, non linear mapping is performed to compress large values and is given by

$$I(x,y) = \beta \tanh\left(\frac{I(x,y)}{\beta}\right). \tag{3.22}$$

The parameters used for normalization techniques are as follows: $\gamma = 0.2$, $\alpha = 0.1$, $\beta = 10$, $\sigma_L = 0.5$ and $\sigma_H = 2$. Result after applying Tan Triggs method is shown in Fig. 3.2(f) for *shaking* video frames 58 and 59. It is observed that bright areas are removed and main details of an image are enhanced.

3.1.7 Difference of Gaussian (DoG)

Difference of Gaussian filtered image is obtained when band pass filter is applied to input image in the logarithmic domain (Struc, 2012). Let $G_{\sigma 1}(x,y)$ represents low pass Gaussian filter with standard deviation σ_1, i.e.,

$$G_{\sigma 1}(x,y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}}. \tag{3.23}$$

Let $G_{\sigma 2}(x,y)$ represents low pass Gaussian filter with standard deviation σ_2, i.e.,

$$G_{\sigma 2}(x,y) = \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}. \tag{3.24}$$

The DoG filtered output is obtained by convolving the input image with $G_{\sigma 1} - G_{\sigma 2}$. $\sigma_1 = 1$ and $\sigma_2 = 2$ are used to select the range of band pass filter. Result of applying

DoG filter on *shaking* video frames is shown in Fig. 3.2(g). DoG filter removes low frequency illumination while retaining high frequency edge details.

3.1.8 Weber Face normalization technique (WF)

Wang *et al.* (2011) proposed an illumination normalization technique impressed by Weber's law. Initially, Gaussian filter smoothens an image $I(x, y)$, followed by the application of Weber Local Descriptor (WLD) (Chen *et al.*, 2010). WLD has two components namely, differential excitation to capture magnitude and orientation direction of intensity variation. According to Lambertian reflectance model, any 2D image is expressed as $I(x, y) = R(x, y)L(x, y)$, where $R(x, y)$ represents reflectance component (depends on object properties) and $L(x, y)$ denotes the illumination component (depends on lighting source). WLD is applied to image to obtain illumination insensitive representation of I, which is named as weber face given by

$$WF(x, y) = \tan^{-1}\left(\alpha \sum_{i=-1}^{1}\sum_{j=-1}^{1} \frac{I(x, y) - I(x - i\Delta x, y - i\Delta y}{I(x, y)}\right). \tag{3.25}$$

Assuming that illumination varies very slowly, above equation leads to

$$I(x - i\Delta x, y - i\Delta y) = R(x - i\Delta x, y - i\Delta y) \times L(x - i\Delta x, y - i\Delta y). \tag{3.26}$$

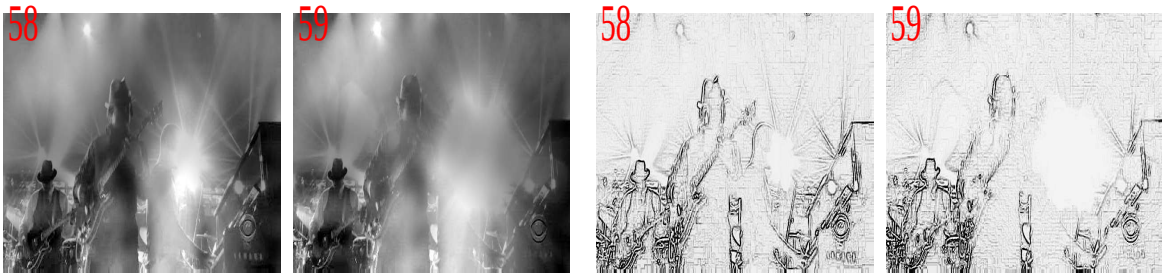
But,

$$L(x - i\Delta x, y - i\Delta y) \approx L(x, y). \tag{3.27}$$

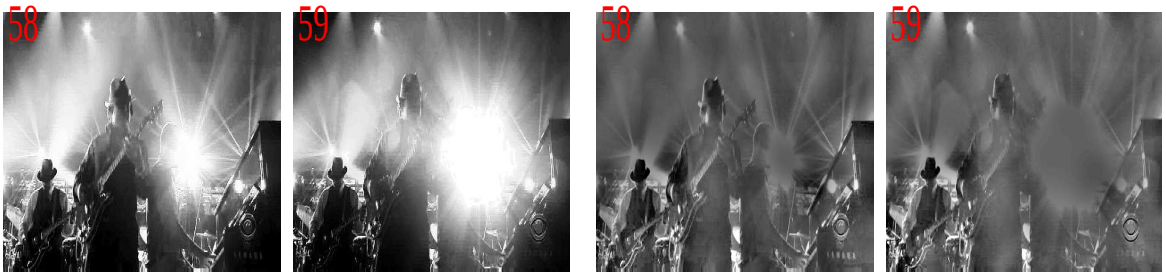
Thus, Weber Face is written as,

$$WF(x, y) = \tan^{-1}\left(\alpha \sum_{i=-1}^{1}\sum_{j=-1}^{1} \frac{R(x, y) - R(x - i\Delta x, y - i\Delta y}{R(x, y)}\right). \tag{3.28}$$

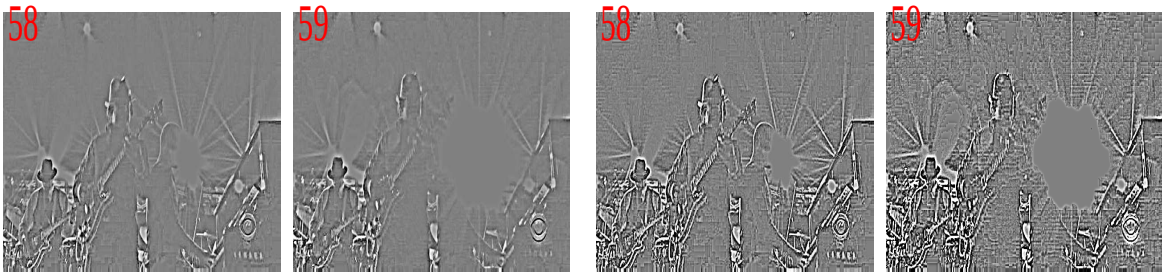
The Weber Face depends only on reflectance component R. Hence this model computes illumination insensitive image which does not depend on illumination factor.



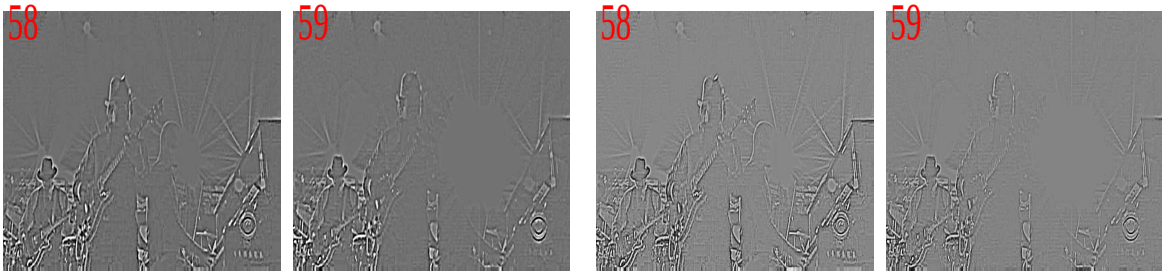
(a) Single Scale Retinex **(b)** Single scale Self Quotient



(c) wavelet based normalization **(d)** Discrete Cosine Transform



(e) Retina Model **(f)** Tan Trigs



(g) Difference of Gaussian **(h)** Weber Face

Figure 3.2: *shaking* frames 58 and 59 after applying photometric normalization techniques

3.2 Illumination invariant median flow tracker

In this section, we review median flow tracker for tracking an object in RGB imagery. In addition, we employ illumination invariant techniques to overcome drift during sudden light changes.

3.2.1 Median Flow Tracker

Median flow tracker (Kalal *et al.*, 2010) has been proposed to track an object based on point tracking using NCC and Forward-Backward (FB) error measure. The tracker accepts bounding box around the object and a pair of image frames as inputs. Let I_t and I_{t+1} denote a pair of image frames, let B_t denotes bounding box around the object in frame t, let B_{t+1} denotes the bounding box obtained as output from median flow tracker. A non overlapping grid of 10×10 points are initialized inside the bounding box, and each point is tracked using Lucas-Kanade (LK) optical flow (Baker and Matthews, 2004) tracker to estimate the motion of object from frame I_t to I_{t+1}. Each point is assigned with an error that includes NCC and FB error. NCC error is calculated based on pixel differences, while FB error is obtained based on Euclidean distance between forward and backward trajectories. LK tracker is utilized to estimate the forward and backward trajectories, which are the set of tracked points from frames I_t to I_{t+k} and I_{t+k} to I_t respectively. The motion of object is predicted using the best points by removing the outliers based on error. To estimate the scale, ratio of point distance in the present frame and that of previous frame is calculated. The median of scale change for set of points is used to predict the scale change in every frame. Thus, median flow tracker achieves greater tracking results in several challenging aspects. However, the accuracy of tracker reduces for sudden illumination variation, fast motion, and occlusion. In this chapter, we address only sudden illumination change issue in detail. Accordingly, image frames are processed using illumination invariant techniques before tracking. The modified median flow tracker is depicted in Fig. 3.3

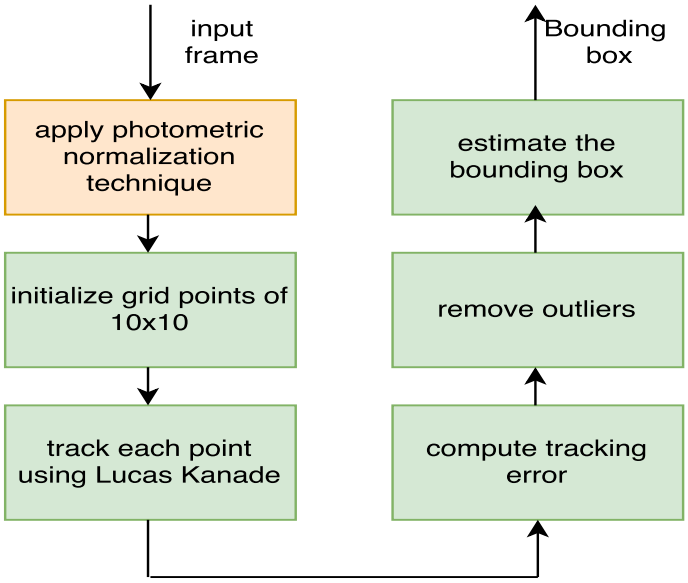


Figure 3.3: Enhanced median flow tracker

3.2.2 Experimental results and analysis

3.2.2.1 Setup

The proposed algorithm is implemented using OPENCV 3.2 and MATLAB 15a software in a machine with intel(R) core i5-5200U, CPU at 2.20GHz processor with 8GB RAM. We make use of INface toolbox (Struc and Pavesic, 2011) for our implementation.

3.2.2.2 Datasets

The modified tracker is evaluated using 5 image sequences chosen from OTB dataset (Wu *et al.*, 2013) possessing sudden illumination variation as a challenging aspect. Since, the enhanced median flow tracker is designed to solve sudden changes in illumination precisely, and it does not affect the performance of baseline tracker in other challenging aspects. Thus, we have selected five challenging videos that have sudden illumination variation at several frames. The videos include *shaking*, *singer2*, *trellis*, *car24* and *man*. Table 3.2 provides the details of video sequences used for experiments.

Table 3.2: The challenges associated with video sequences from Object Tracking Benchmark dataset.

Sequence	Frame Size	No of frames	illumination details	environment
shaking	352 ×624	365	flash light effects within and between frames	indoor flash light
singer2	352 ×624	366	flash light effects between frames	indoor flash light
man	193 ×241	134	light change between frames	indoor lights
trellis	240 ×320	569	light change within frame	indoor shadow
car24	240 ×320	3059	light change between frames	outdoor shadow

3.2.2.3 Qualitative analysis

In Fig. 3.4(a), *man* video is presented, in which a person walks inside the dark room and is suddenly lighted. MFT slightly drifts off due to abrupt changes in illumination. However, modified MFT is more robust for light changes and tracks the person all through the image sequences. In the proposed tracker, light variation is compensated using photometric normalization techniques. Thus, retina model based MFT (MFT_RM) produces high precision and success rate compared to other techniques. Also, recent famous trackers such as DFT, EDFT, and corrected background weighted histogram (CBWH) fail to track the complete sequence. Fig. 3.4(b) depicts *shaking* video, which shows a person (target) singing in a music concert program. The video has flashlight effects causing sudden local/global intensity variation. Thus, local illumination change around the target area confuses tracker due to uneven light scattering. MFT uses optical flow method to find the location of object, which is sensitive to rapid intensity variation. In the proposed work, modified tracker MFT_RM is committed to reject the illumination effects. Consequently, modified tracker tracks singer in *shaking* sequence until last frame. Global illumination change is observed during transition from 58^{th} to 59^{th} frame due to flashlight effects. Such changes are encountered frequently, which makes MFT and other well-known trackers to lose track. However, modified MFT is competent to handle sudden light changes. Among the selected trackers, circulant structure kernel (CSK) tracker can track whole sequence, while rest of the trackers fail during light changes.

In Fig. 3.4(c), *singer2* video frames are displayed, which has back-light effects due to which many trackers cannot track. However, the proposed tracker do not fail due to an unexpected change in illumination, however misses the target due to heavy pose variation.

In Fig. 3.4(d), *car24* sequence is illustrated, baseline MFT tracker loses the target when car moves from shadow region to sunlight area. But, the proposed tracker tracks the sequence till end. Some trackers such as DAT, EDFT also perform satisfactorily in this video. TLD also shows promising results in *car24* video due to the presence of re-detection unit. It can re-detect the object even when tracker misses the target due to illumination changes.

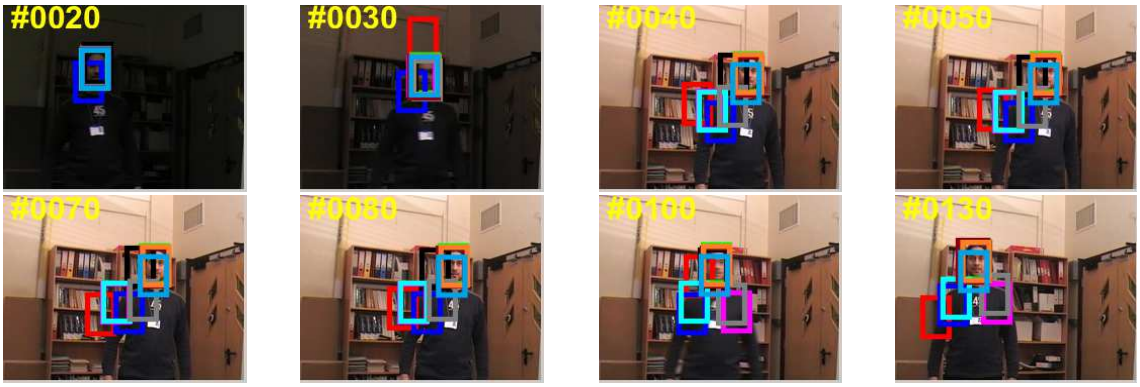
In Fig. 3.4(e), *trellis* sequence, a person walks under a trellis, and affected by the local illumination variation due to shadows. As a result, MFT tracker slowly drifts away from the target. However, photometric normalization technique is able to remove illumination related problems. Hence, enhanced MFT improves the base tracker to a large extent by effectively handling local and global illumination changes.

The center location error plots are shown in Fig. 3.5. The graph corresponding to modified median flow tracker is close to x-axis.

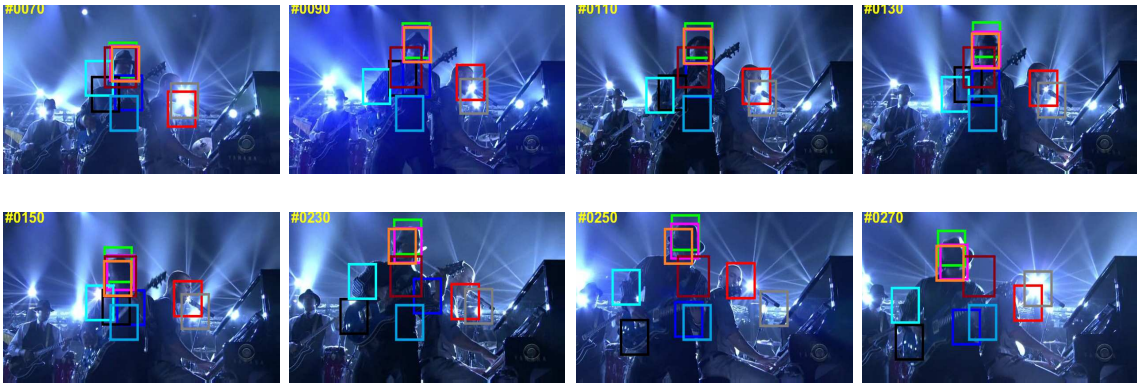
3.2.2.4 Quantitative analysis

To evaluate the trackers, distance precision score and overlap precision score are utilized. In addition, precision plot and success plots are employed to rank the trackers pictorially as a part of quantitative assessment.

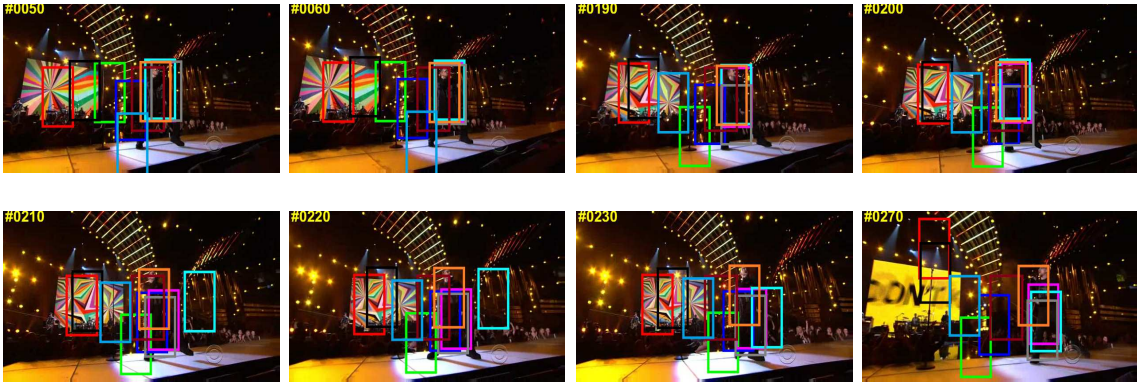
Even though all the illumination normalization techniques are intended to solve sudden/gradual and local/global illumination challenges, majority of them are not acceptable for visual tracking experiments. The practical reasons include (i) they fail to attain the speed required for processing (ii) they are not effective to derive the reflectance part. For video tracking application, illumination component needs to be removed or maintained consistent all over the frames.



(a) *man*



(b) *shaking*



(c) *singer2*

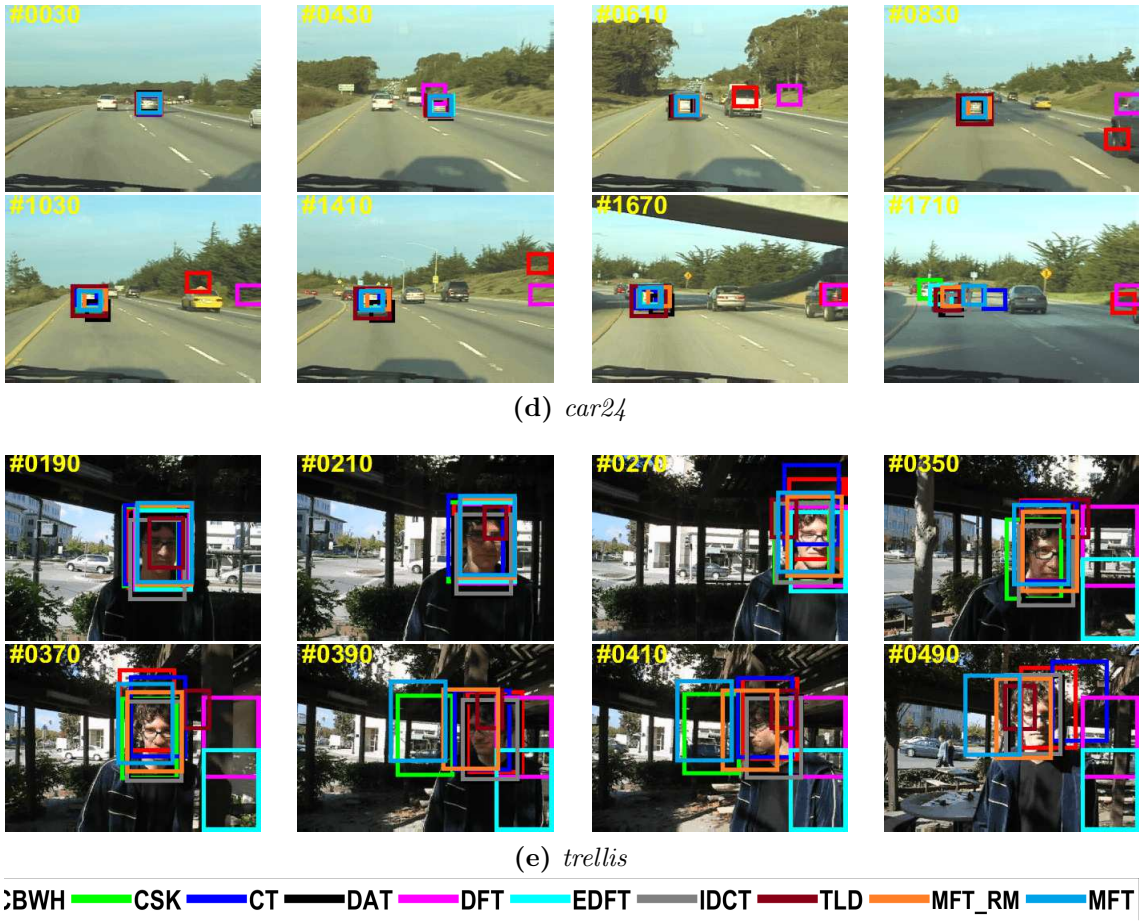
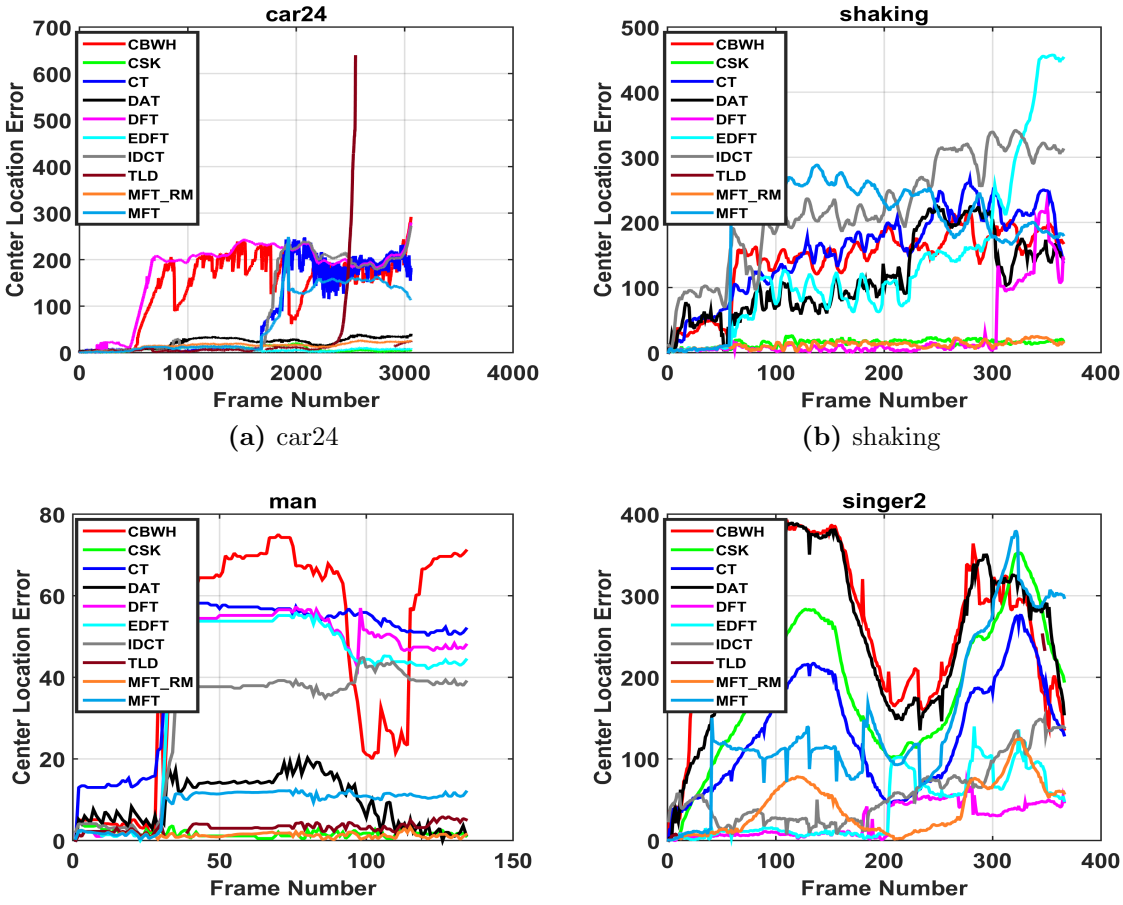


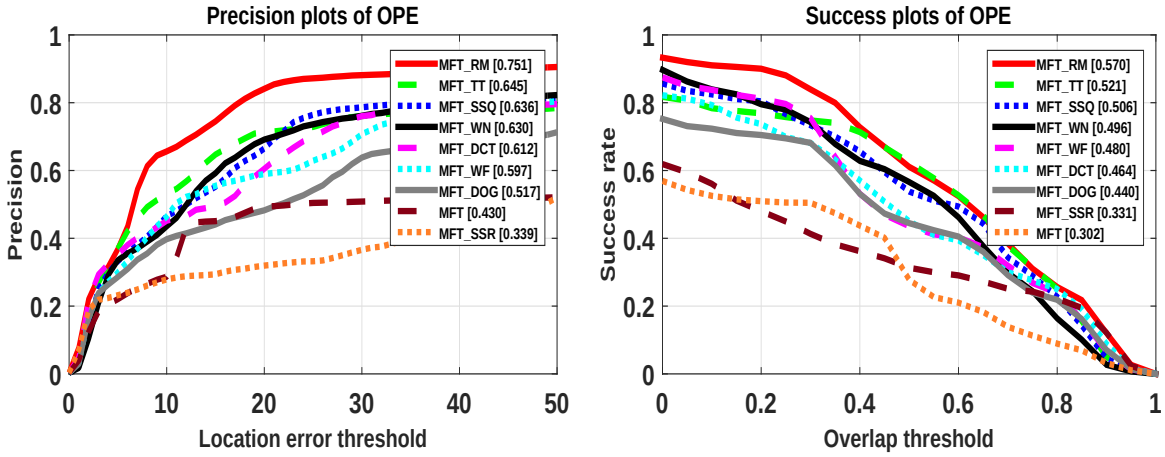
Figure 3.4: Qualitative analysis of the modified tracker and state-of-the-art trackers on image sequences *man*, *shaking*, *singer2*, *car24*, and *trellis*.



(a) car24 **(b)** shaking

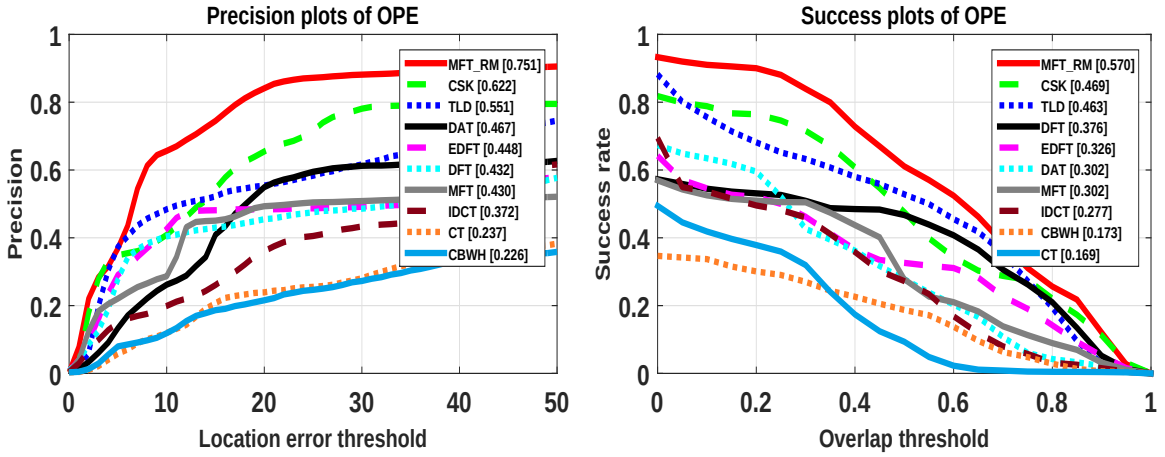
(c) man **(d)** singer2

Figure 3.5: Center location error plots of state-of-the-art trackers on image sequences posing rapid illumination change as a challenging aspect.



(a) precision plot **(b)** Success plot

Figure 3.6: Precision plot and success plots of median flow tracker and its modified versions on five sudden illumination changing videos



(a) precision plot **(b)** success plot

Figure 3.7: Precision plot and success plots of the modified median flow tracker (MFT_RM) and state-of-the-art trackers tested on five illumination challenging sequences

Table 3.3: Distance precision score of median flow tracker (MFT) and modified median flow tracker on five challenging video sequences with abrupt light changes. The best results are displayed in boldface

	MFT	MFT_DCT	MFT_DOG	MFT_RM	MFT_SSQ	MFT_SSR	MFT_TT	MFT_WF	MFT_WN
shaking	15.61	89.31	49.86	**100**	98.90	15.89	100	41.91	99.72
singer2	10.92	10.10	16.66	**48.63**	42.07	23.49	44.80	19.12	25.40
man	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
trellis	70.82	68.89	88.22	87.34	83.47	10.19	91.03	**95.07**	75.57
car24	54.82	81.75	16.31	**100**	57.82	16.54	28.44	58.25	67.40
mean	50.43	70.01	54.21	**87.19**	76.45	33.22	72.85	62.87	73.62

Table 3.4: Overlap precision score of median flow tracker (MFT) and improved median flow tracker on five challenging video sequences with abrupt light changes. The best score is displayed in boldface.

	MFT	MFT_DCT	MFT_DOG	MFT_RM	MFT_SSQ	MFT_SSR	MFT_TT	MFT_WF	MFT_WN
shaking	15.61	50.41	29.31	81.36	48.49	15.89	**88.21**	22.46	78.63
singer2	10.92	9.01	14.75	**41.25**	38.52	25.13	39.34	17.48	23.49
man	36.56	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
trellis	58.52	48.50	65.55	65.55	64.67	4.92	**67.13**	65.90	63.62
car24	17.26	17.22	12.71	**17.26**	17.26	10.62	17.26	11.96	17.26
mean	27.77	45.03	44.46	61.08	53.79	31.31	**62.39**	43.56	56.60

Table 3.5: Distance precision score of the state-of-the-art trackers tested on video sequences with sudden illumination change as a challenge. The modified median flow tracker shows improved average distance precision score as compared to the baseline tracker.

	CBWH	**CSK**	**CT**	**DAT**	**DFT**	**EDFT**	**IDCT**	**TLD**	**MFT_RM**	**MFT**
shaking	2.73	76.98	4.38	3.28	83.01	16.71	1.36	43.28	**100**	15.61
singer2	0.27	4.09	0.81	1.63	**62.84**	56.01	28.68	13.66	48.63	10.92
man	29.10	**100**	22.38	**100**	23.13	23.13	25.37	**100**	**100**	**100**
trellis	73.63	86.29	45.16	**92.44**	52.72	47.62	91.21	49.73	87.34	70.82
car24	17.48	90.35	54.88	**100**	15.92	**100**	56.03	84.89	**100**	54.82
mean	24.64	71.54	25.52	59.47	47.52	48.69	40.53	58.31	**87.19**	50.43

Table 3.6: Overlap precision score of the state-of-the-art trackers tested on video sequences with sudden illumination change as a challenge. The modified median flow tracker (MFT_RM) shows improved mean overlap precision score when compared to the baseline tracker.

	CBWH	**CSK**	**CT**	**DAT**	**DFT**	**EDFT**	**IDCT**	**TLD**	**MFT_RM**	**MFT**
shaking	1.64	58.08	4.10	3.01	**82.46**	16.16	1.09	39.45	81.36	15.61
singer2	0.27	3.55	1.36	1.36	**69.67**	59.56	25.95	13.38	41.25	10.92
man	20.89	**100**	0.74	47.01	22.38	22.38	23.13	95.52	**100**	36.56
trellis	55.36	59.05	27.06	71.00	51.84	47.62	**68.71**	42.00	65.55	58.52
car24	15.39	17.26	13.40	16.44	7.19	17.26	17.26	**73.78**	17.26	17.26
mean	18.71	47.58	9.33	27.76	46.71	32.60	27.23	52.82	**61.08**	27.77

We provide the performance comparison of modified median flow tracker (a combination of different illumination normalization techniques and median flow tracker) with the baseline median flow tracker through the precision score and overlap score performance metrics. Thus, Table 3.3 and Table 3.4 respectively tabulates DP and OP scores of modified median flow tracker. It is observed from experiments that MFT_RM, Tan Triggs (MFT_TT) and Single scale Self Quotient image (MFT_SSQ) are found to be more efficient for tracking of an object in a video which has global and local illumination variations. However, single scale retinex (MFT_SSR) performs poorly on videos with unexpected illumination changes. The modified median flow tracker is intended to overcome drift due to variations of illumination related problems. However, many causes such as pose variation, rotation, blur, scale, etc. are not discussed in this part of the work.

MFT_RM performs best among other normalization techniques in real-time, hence it is used to compare with the recent state-of-the-art trackers. The compared trackers include CBWH (Ning *et al.*, 2012), CSK (Henriques *et al.*, 2012), CT (Zhang *et al.*, 2012), DAT (Possegger *et al.*, 2015), DFT (Sevilla-Lara and Learned-Miller, 2012), EDFT (Felsberg, 2013), IDCT (Asvadi *et al.*, 2013), TLD (Kalal *et al.*, 2012) and baseline MFT (Kalal *et al.*, 2010). On an average, MFT_RM outperforms the compared trackers in terms of average precision score and success scores as tabulated in Table 3.5 and Table 3.6 respectively. In addition, Fig. 3.7(a) and Fig. 3.7(b) depict the precision plots and success plots for one pass evaluation.

3.3 Illumination consistent median flow tracker

In this approach, illumination is maintained constant throughout the video irrespective of light variations.

3.3.1 Illumination constancy using DCT

Section 3.1 explained the acquisition of an image based on retinex theory. In the previous work, several photometric normalization techniques have been studied to

nullify the illumination induced problems. In this section, we alter discrete cosine transform (DCT) coefficients to maintain constant illumination. Thus, from Eq. (3.6), illumination difference is obtained as;

$$\varepsilon = \ln I_{t+1}(x, y) - \ln I_t(x, y) \tag{3.29}$$

Therefore, the difference between two images provide desired change in illumination which needs to be adjusted to maintain constant intensity over video frames.

3.3.1.1 Modification of DCT coefficients

DCT is used in signal processing area for several applications such as compression due to its high energy compaction property. In the proposed work, we exploit DCT domain to compensate illumination variations by adding or subtracting the additive term ε in the logarithmic domain. The illumination is considered to be slowly varying component in an image compared to the reflectance component. Thus, the low-frequency coefficients of DCT domain are directly related to illumination component. In addition, DC coefficient of DCT domain decides the average illumination of image. Therefore, the expected uniform illumination can be attained by fixing the DC coefficient to uniform value in every frame. To achieve this, running weighted sum of frames is computed with an adaptation rate $\eta = 0.01$. However, very low and highly illuminated (poor contrast in both cases) video frames are omitted from an average calculation. We utilize entropy property of an image to estimate its contrast. Typically, entropy value falls down for less contrast image frames (which is the result of low illumination or high illumination). Entropy of an image is calculated as

$$E = -\sum_k p(k) \log_2 p(k), \tag{3.30}$$

where $p(k)$ denotes probability density function of image pixels. Hence, current frame is used to determine the average image only if the normalized entropy difference between present and previous frame differs by a threshold of θ, i.e.

$$|E(f) - E(f-1)| < \theta \tag{3.31}$$

where f denotes the frame number. In the proposed scheme, θ is set at 5. Therefore, an average image which provides global illumination is calculated using,

$$s(x, y) = s(x, y) + \eta \; I_f(x, y) \tag{3.32}$$

Mean value of the average image in logarithmic domain is calculated as,

$$meanS = \frac{1}{M * N} \sum_x \sum_y ln \; s(x, y) \tag{3.33}$$

Therefore, the desired uniform illumination can be realized by fixing DC coefficient of every frame to $meanS \times \sqrt{(M \times N)}$, where (M, N) denotes the size of each frame. The low-frequency DCT coefficients are highly associated with illumination and need to be discarded to compensate for illumination differences. Thus, upper left corner of DCT image denotes low-frequency coefficients which are set to zero by ordering the coefficients in a zig-zag fashion. If less number of coefficients are discarded, then the effect of illumination is still present in the image. Similarly, neglecting large number of coefficients results in removal of features. Hence, 25 coefficients are replaced by zero to eliminate the light changes and retain texture features. Also, DC coefficient in DCT domain is replaced by $meanS \times \sqrt{(M \times N)}$ as given by Eq. (3.33), which helps to sustain the consistent illumination. Finally, inverse DCT is applied to get the normalized image frame with constant intensity throughout the video. The entropy of DCT normalized image remains almost uniform compared to that of original image as depicted in Fig. 3.8. Fig. 3.9 details about how illumination effects are nullified using the proposed method.

3.3.1.2 Video enhancement

After normalizing each frame, contrast of an image is increased using adaptive gamma enhancement technique (Huang *et al.*, 2013). Let x denote the pixel value of a normalized image. The adaptive gamma correction uses $F_X(\mathbf{x})$ (cumulative density function) as an adaptive parameter to compute gamma. Point transformation of each pixel value is defined as,

$$T(x) = x_{\max} \left(\frac{x}{x_{\max}} \right)^{\gamma}, \tag{3.34}$$

where x_{max} denotes maximum value of x . Weighting distribution function is calculated to modify the histogram $f_X(x)$ of an image as follows:

$$f_X(x) = {f_X}^{\max}(x)\left(\frac{f_X(x) - {f_X}^{\min}(x)}{{f_X}^{\max}(x) - {f_X}^{\min}(x)}\right)^{\alpha}, \tag{3.35}$$

where the parameter α is empirically fixed at 0.6, as it provides the better contrast of an image in the proposed tracking method. $f_X(x)$ represents the histogram of an image with maximum of ${f_X}^{\max}(x)$ and minimum of ${f_X}^{\min}(x)$. Cumulative density function $F_X(x)$ is obtained as given below,

$$F_X(x) = \sum_{x=0}^{255}\left(\frac{f_X(x)}{\sum_{0}^{255} f_X(x)}\right), \tag{3.36}$$

Adaptive parameter γ is computed using $F_X(x)$ to enhance the video frame according to Eq. 3.34. i.e.,

$$\gamma = 1 - F_X(x) \tag{3.37}$$

The image contrast enhancement helps to improve the tracking accuracy. This process is repeated for every frame before tracking process. Thus, the proposed algorithm for tracking using DCT based illumination normalization, gamma enhancement, and median flow tracker is summarized in Algorithm 1.

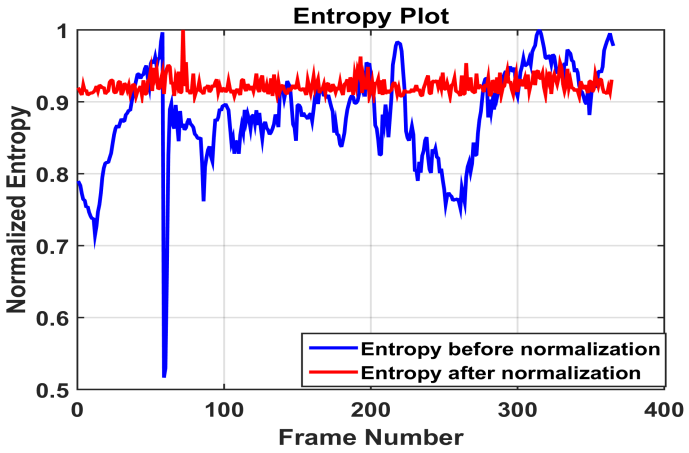
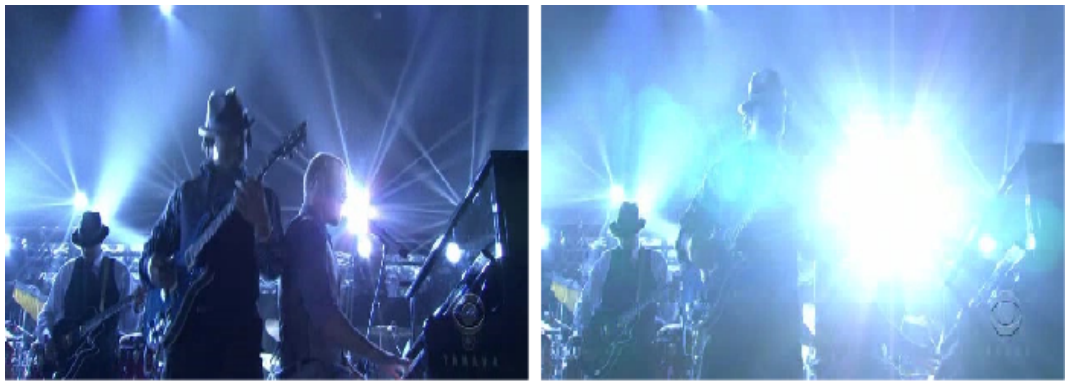
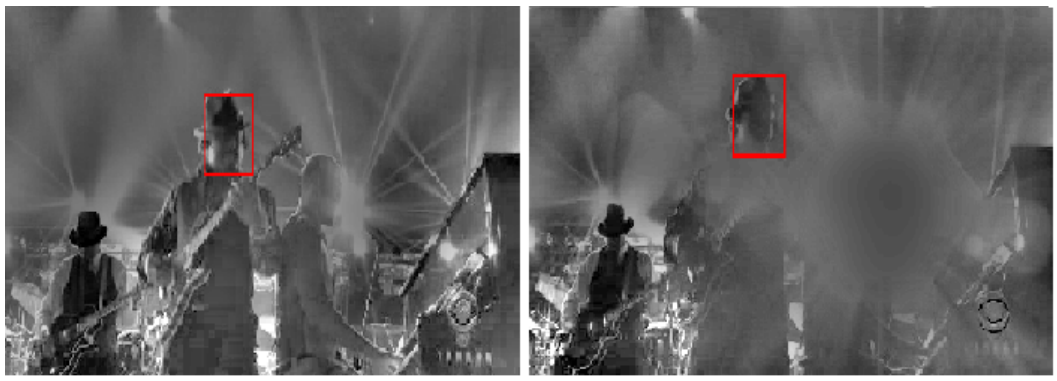


Figure 3.8: The entropy modification using DCT based illumination normalization technique.



(a)

(b)



(c)

Figure 3.9: a) original 58^{th} and 59^{th} frames of *shaking* video b) gray scale images of 58^{th} and 59^{th} frames (c) illumination normalized images after modifying the DCT coefficients of 58^{th} and 59^{th} frames.

Algorithm 1 Proposed illumination constancy algorithm for median flow tracker

Require: Image frames $I_1, I_2, ...I_n$, ground-truth bounding box $B_1 = [x, y, w, h]$ of the object in the first frame I_1.
Return: Bounding box $B_2, B_3....B_n$
for $f = 1$ **to** n **do**
  if $f > 1$ **then**
    calculate entropy $E(f)$ as in Eq. (3.30)
    if $|E(f) - E(f-1)| < \theta$ **then**
      $s(x, y) = s(x, y) + \eta\ I_f(x, y)$
      $meanS = \frac{1}{M*N} \sum_x \sum_y ln\ s(x, y)$
    end if
    find $C_f\,(u, v) = dct2(I_f(x, y))$
    Set $C_f(0, 0) = meanS \times \sqrt{M * N}$
    Arrange $C_f(u, v)$ in a zig-zag fashion and make first 25 elements zero except DC coefficient.
    Find $I_f(x, y) = idct2(C_f(u, v))$.
    Apply adaptive gamma enhancement algorithm on $I_f(x, y)$.
    Find bounding box B_f using median flow tracker as given in section 3.2.
  end if
end for

3.3.1.3 Experimental results and analysis

In this section, we provide the details of videos, quantitative and qualitative analysis to demonstrate the effectiveness of the proposed tracker. The proposed tracker has been implemented and evaluated on five challenging videos. We have selected the videos from OTB dataset which poses sudden illumination change at many places. The videos are *shaking*, *singer2*, *trellis*, *car24*, and *man*.

Table 3.7: Distance Precision (DP) score and Overlap Precision (OP) score of state-of-the-art trackers.

	CBWH	**CSK**	**CT**	**DAT**	**DFT**	**EDFT**	**IDCT**	**TLD**	**IVMFT**	**MFT**
DP	24.65	71.55	25.47	59.47	47.53	48.70	40.54	58.31	**77.43**	50.44
OP	18.71	47.59	9.12	27.77	46.71	32.60	27.23	52.82	**53.13**	27.78

Table 3.8: Comparison of distance precision score of the proposed tracker (IVMFT) with recent trackers

	CBWH	CSK	CT	DAT	DFT	EDFT	IDCT	TLD	IVMFT	MFT
shaking	2.74	76.99	4.11	3.29	83.01	16.71	1.37	43.28	100.00	15.62
singer2	0.27	4.10	0.82	1.64	62.84	56.01	28.69	13.66	21.58	10.93
man	29.10	100.00	22.39	100.00	23.13	23.13	25.37	100.00	100.00	100.00
trellis	73.64	86.29	45.17	92.44	52.72	47.63	91.21	49.74	65.55	70.83
car24	17.49	90.36	54.89	100.00	15.92	100.00	56.03	84.90	100.00	54.82

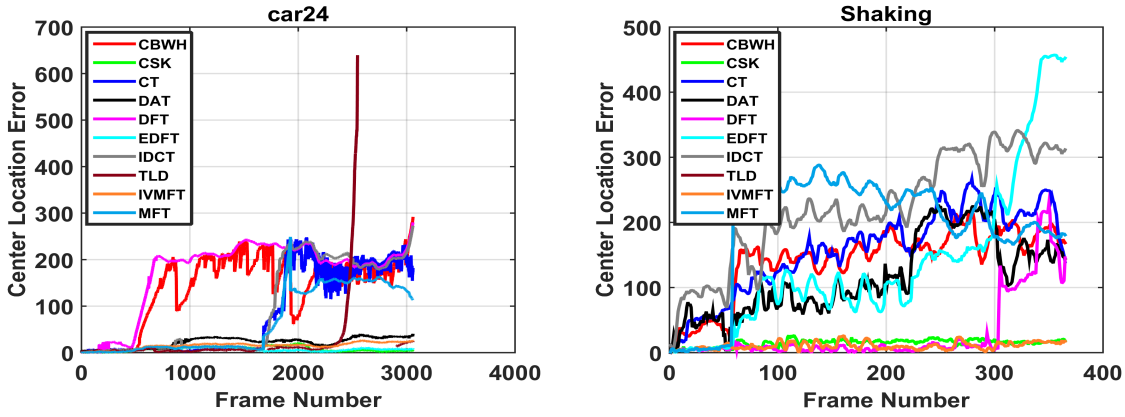
Table 3.9: Comparison of overlap precision score of the proposed tracker (IVMFT) with recent trackers

	CBWH	CSK	CT	DAT	DFT	EDFT	IDCT	TLD	IVMFT	MFT
shaking	1.64	58.08	3.01	3.01	82.47	16.16	1.10	39.45	82.47	15.62
singer2	0.27	3.55	1.37	1.37	69.67	59.56	25.96	13.39	20.22	10.93
man	20.90	100.00	0.75	47.01	22.39	22.39	23.13	95.52	100.00	36.57
trellis	55.36	59.05	27.07	71.00	51.85	47.63	68.72	42.00	45.69	58.52
car24	15.40	17.26	13.40	16.44	7.19	17.26	17.26	73.78	17.26	17.26

The proposed illumination invariant median flow tracker is named as IVMFT and is compared with the recent tracking methods which are more popular in video tracking field like CBWH (Ning *et al.*, 2012), DAT (Possegger *et al.*, 2015), CT (Zhang *et al.*, 2012), IDCT (Asvadi *et al.*, 2013), EDFT (Felsberg, 2013), TLD (Kalal *et al.*, 2012), DFT (Sevilla-Lara and Learned-Miller, 2012), CSK (Henriques *et al.*, 2012). From experimental analysis, IVMFT performs best compared to existing trackers in terms of distance precision score and overlap precision score as shown in Table 3.7 and 3.8. Furthermore, precision plots and success plots of state-of-the art trackers are depicted in Fig. 3.12.

3.3.1.4 Qualitative analysis

In Fig. 3.11(a). *man* video is presented, in which a person moves in a dark room and is suddenly illuminated. IVMFT is successful in tracking the object throughout the sequence irrespective of illumination changes. MFT loses track due to sudden changes in light, whereas the proposed tracker is made illumination independent to track all the way through sequences. However, recent trackers such as DFT, EDFT, CBWH fail to track complete sequence.



(a) car24 **(b)** shaking

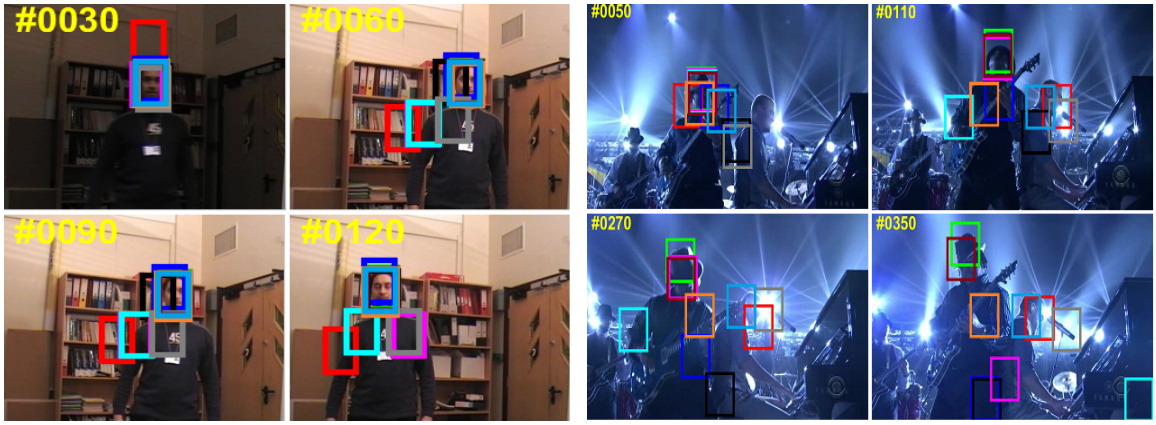
(c) man **(d)** singer2

Figure 3.10: Center location error plots of state-of-the-art trackers on image sequences posing heavy illumination change as a challenging aspect.

Fig. 3.11(b) depicts shaking video, which shows a person (target) singing in a music program with flash light effects. It can be observed that the light changes are very sudden. Since the base tracker is designed using optical flow technique, it is sensitive to intensity variations. Hence, the modified tracker IVMFT is designed to maintain constant illumination to track the person in shaking sequence till video completion. Flash light effects are observed in many frames such as 59^{th} and 60^{th}, where median flow tracker fails to track, however, the proposed tracker is very robust in handling such problems. Also, CSK tracker is able to track the complete sequence, whereas rest of the trackers fail due to illumination changes.

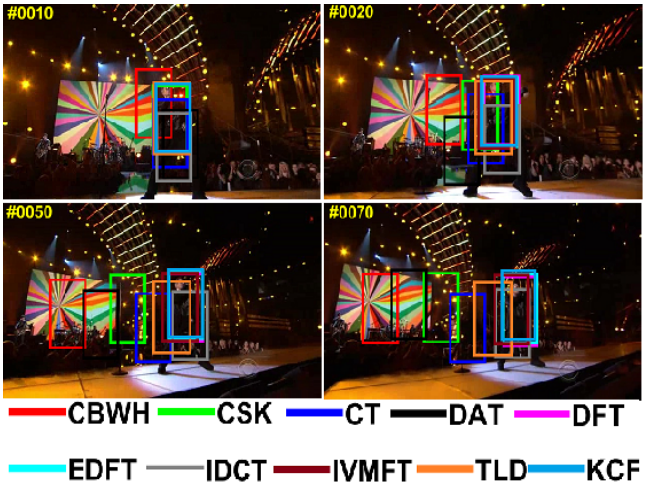
In Fig. 3.11(c). *singer2* video frames are presented, which has more backlight

effects due to which many trackers cannot track successfully. However, the proposed tracker do not fail due to an unexpected change in illumination, but it misses the target due to heavy pose variation



(a) man

(b) shaking



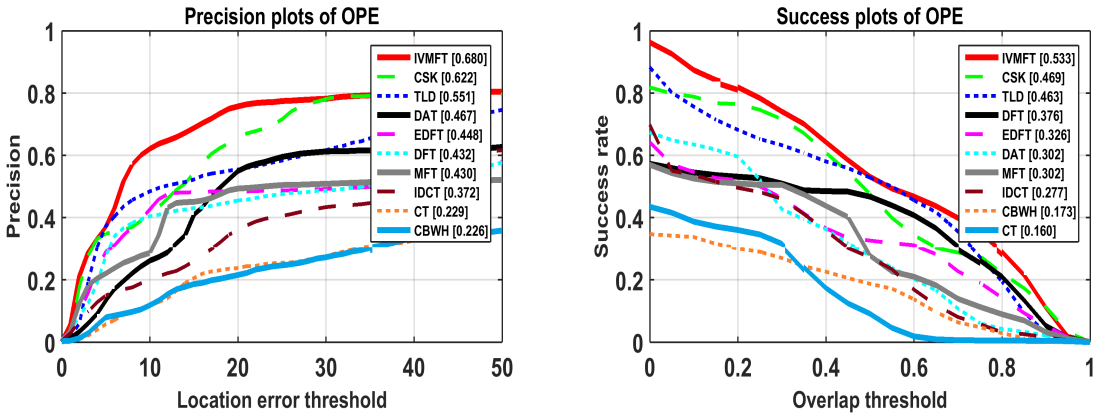
(c) singer2

Figure 3.11: Qualitative evaluation of IVMFT with state-of-the-art tracker for ***man***, ***shaking***, and ***singer2*** sequences.

3.3.1.5 Quantitative analysis

The proposed IVMFT is compared quantitatively with other trackers in terms of distance precision score and overlap precision score. Table 3.7 shows mean DP and OP score of state-of-the-art trackers. IVMFT performs best among other selected trackers

for five challenging video sequences. Fig. 3.12 depicts precision plot and success plots and proposed tracker tops among all.



(a) precision plot **(b)** success plot

Figure 3.12: The comparison of precision plot and success plots of IVMFT (enhanced MFT) with recent trackers

3.4 Summary

Illumination pre-processing is the most effective method to handle rapid and gradual light changes to mitigate tracking drift for videos with sudden illumination problems. This chapter provided an insight into eight illumination invariant techniques and embedded with well-known median flow tracker. The proposed approach has been evaluated on videos with rapidly varying illumination. The tested image sequences contain indoor and outdoor videos with flash light effects and shadow casting. The proposed modified median flow tracker has been compared with recent trackers, and experimental results show that MFT_RM outperforms favorably on suddenly varying illumination videos. Also, MFT_TT equally performed well in terms of accuracy and speed. The execution time plays crucial stage in real-time tracking which is satisfied by the retina model.

In addition, an efficient approach has been presented to maintain a constant illumination within and between frames of a video. To undertake this issue, DCT coefficients were tailored to eliminate the effect of illumination changes. The contrast

of an image has been improved through adaptive gamma correction technique. Several video sequences with sudden changing illumination have been considered to test the tracker. The experimental results clearly demonstrated that the proposed algorithm improves the median flow tracker for slow/abrupt changes in the illumination.

Chapter 4

TRACKING WITH CONDITIONAL SWITCHING

In this chapter, we address selection of features, adaptive learning rate and switching techniques to minimize tracking drift using kernelized correlation filter based tracker. The correlation filters have been extensively used in object tracking due to its robustness and attractive computational speed. But, the correlation filters are more sensitive to occlusion, fast motion and object deformation because they are trained using spatial features. Besides, updating the filter template with slightly drifted or occluded samples increase the probability of tracking failure. In contrast, the median flow tracker is complementary to correlation techniques and is fast, robust to occlusion and deformation, but sensitive to illumination variation. In this chapter, we utilize the advantage of correlation and optical flow based trackers to achieve drift free tracking. In this work, we apply the correlation filter based tracker to track an object and switch to the modified median flow tracker during drift conditions. The combined model is optimized to cope up with fast appearance changes and overcome from drifting. In addition, we present an adaptive feature selection process to select the most discriminative feature/features among color name and histogram of oriented gradient features based on object separation from background in intensity and color channels. The proposed tracker updates the filter template dynamically, depending on appearance of an object using adaptive learning rate to track the target irrespective of occlusion, motion blur, and deformation. The scale of object is estimated using Lu-

cas Kanade homography method. The experiments are carried out using challenging video sequences from a standard OTB dataset and show the best performance among baseline trackers.

This chapter is organized as follows. The section 4.1 reviews the famous KCF tracker. Section 4.2 presents the proposed switching technique and section 4.3 summarizes the work.

4.1 Background

4.1.1 KCF tracker

The prime focus of this work is to improve KCF (Henriques *et al.*, 2015) tracker; it is hence presented in this section. KCF is a kernelized version of correlation filter that uses dense periodic sampling to generate the set of positive and negative samples. Thus, the model exploits circulant structure of the sample image. The feature space is constructed for an image patch $\mathbf{x_i}$ of size $P \times Q$ in a given frame. All circular shifts of the input sample are used to train the correlation filter to produce the output $\mathbf{y}$ of same size. The desired response $\mathbf{y}$ is considered to have Gaussian shape with the maximum value at the center. A patch size of 1.5 times the object is cropped to avoid spectrum aliasing due to circular convolution. The sharp boundaries are smoothed by multiplying the patch with cosine window. Thus, the training problem is formulated to learn a filter to minimize the error between circular shifts of kernel mapped training sample $\mathbf{x_i}$ and the desired output $\mathbf{y}$ as follows:

$$arg\min_{\mathbf{w}} \sum \left(\langle \phi(\mathbf{x_i}), \mathbf{w} \rangle - \mathbf{y}\right)^2 + \lambda \|\mathbf{w}\|^2, \tag{4.1}$$

where, $\mathbf{w}$ is the filter template in the spatial domain, λ represents regularization term and $\phi(\mathbf{x_i})$ maps the input sample $\mathbf{x_i}$ into Gaussian kernel space. In expanded form, Gaussian kernel space is given by

$$\kappa(\mathbf{x_l}, \mathbf{x_j}) = exp\left(-\frac{1}{\sigma_g^2}\left(\left\|\mathbf{x_l^2}\right\| + \left\|\mathbf{x_j^2}\right\| - 2\Im^{-1}\left(\sum \hat{\mathbf{x}}_\mathbf{l} \odot \hat{\mathbf{x}}_\mathbf{j}^{*}\right)\right)\right) \tag{4.2}$$

where σ_g is the variance of Gaussian kernel function, $\odot$ denotes element-wise multiplication, $*$ denotes complex conjugate of a variable, the symbol $\Im$ denotes Discrete Fourier Transform (DFT) and $\Im^{-1}$ represents Inverse Discrete Fourier Transform (IDFT). The filter template, $\mathbf{w}$ can also be expressed as a linear combination of input data samples as

$$\mathbf{w} = \sum \beta_i \phi(\mathbf{x_i}), \tag{4.3}$$

where $\beta_\mathbf{i}$ is solved as

$$\beta_\mathbf{i} = \Im^{-1}\left(\frac{\Im(\mathbf{y})}{\Im(\kappa(\mathbf{x_i}, \mathbf{x_i})) + \lambda}\right) \tag{4.4}$$

In order to consider appearance changes over time, the template is updated with new samples in the frequency domain using a fixed learning rate η as

$$\hat{\mathbf{x}}_\mathbf{t} = (1-\eta)\,\hat{\mathbf{x}}_{\mathbf{t-1}} + \eta\Im(\mathbf{x_i}), \tag{4.5}$$

$$\hat{\alpha}_\mathbf{t} = (1-\eta)\,\hat{\alpha}_{\mathbf{t-1}} + \eta\Im(\beta_\mathbf{i}), \tag{4.6}$$

where, $\hat{\mathbf{x}}_\mathbf{t}$ denotes the learned template in the frequency domain, $\hat{\alpha}_\mathbf{t}$ denotes the filter template in the frequency domain, $\Im(\mathbf{x_i})$ denotes the training sample in the frequency domain. In the first frame, $\hat{\mathbf{x}}_\mathbf{t} = \Im(\mathbf{x_1})$ and $\hat{\alpha}_\mathbf{t} = \Im(\beta_\mathbf{1})$.

The object is detected in the next frame by applying the filter to a search region cropped from the present frame at a previously obtained location. Thus, for the test sample $\mathbf{z}$, which is cropped from the present frame, feature is extracted and then the kernel function is applied to get $\kappa(\mathbf{x_t}, \mathbf{z})$ using Eq. (4.2). Finally, the output response $\mathbf{o}$ is obtained via convolution between the test sample $\mathbf{z}$ and filter template $\hat{\alpha}_\mathbf{t}$ as

$$\mathbf{o} = \Im^{-1}(\hat{\alpha}_\mathbf{t} \odot \Im(\kappa(\mathbf{x_t}, \mathbf{z}))). \tag{4.7}$$

The difference between present and previous object position is observed to locate the target in present frame as $(x_c, y_c) = \underset{(x_c, y_c)}{\arg\max}(\mathbf{o})$.

4.2 Tracking with conditional switching

The general strategy in multi-approach tracking (Santner *et al.*, 2010) is to run the trackers in parallel and combine the individual outputs, either changing between them

or fusing them with probabilistic weights. In parallel approach, the final response of tracker is determined either by selection or combination method (Bertinetto *et al.*, 2016), and in series approach, other trackers are assessed when the first tracker gives less certainty about tracked location. The multi-approach tracking also suggests combining tracking techniques (Samuel *et al.*, 2017) efficiently. The proposed approach is depicted in Fig. 4.1. It uses cascade approach by utilizing KCF as the base tracker with a weighted confidence of color name and HoG features to locate the object. The weights are computed to select the best features among color or texture feature. The selection of feature relies on how well the target is separable from its surroundings in intensity and color channels. The peak to sidelobe ratio (PSR) is observed in every frame to change over to modified median flow tracker based on pre-defined condition. Also, we employ dynamic learning rate to update the filter model of kernelized correlation filter to adapt for the appearance variations and occlusions.

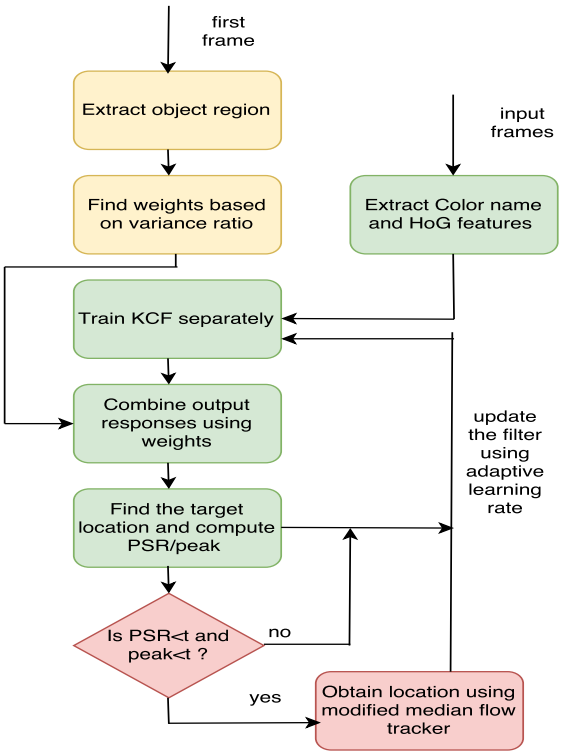


Figure 4.1: General block diagram of the proposed approach

4.2.1 Feature selection

Van *et al.* (2009) discussed the mapping of RGB values to color names for the real-world applications. The colors are extracted from a set of basic colors known as linguistic color labels. The English language has 11 primary colors; they are black, blue, brown, gray, green, orange, pink, purple, red, white and yellow. The color name feature (**CN**) of an image is a 3D vector that contains the probability of color names for each pixel in an image (Khan *et al.*, 2012) and is represented as

$$\mathbf{CN}= \{\mathrm{p}(\mathbf{cn_1}\,|\mathbf{I}\,),....\mathrm{p}(\mathbf{cn_{11}}\,|\mathbf{I}\,)\} \tag{4.8}$$

with

$$p(\mathbf{cn_i}|\mathbf{I}) = \frac{1}{N}\sum_{l\in I} p(\mathbf{cn_i}|f(l)), \tag{4.9}$$

where $\mathbf{cn_i}$ denotes i^{th} color name, l represents the spatial coordinates, N denotes the total number of pixels in an image I, and f represents $L*a*b$ values. Thus, $p(\mathbf{cn_i}|\mathbf{f})$ is obtained using Bayes law by assuming equal priory to all the color names. The color name feature is robust to motion blur, and hence it is used as one of the feature channels to represent the color of an object. HoG features have been introduced in the field of object detection (Dalal *et al.*, 2005), and its variant known as fHoG is used to extract the texture features of an object (Felzenszwalb *et al.*, 2010). A cell-based feature map is calculated by using nine contrast-insensitive orientations and 18 contrast-sensitive orientations. Thus, HoG feature is invariant to illumination and deformation, hence the proposed method makes use of 27 HoG feature channels to represent the texture features of an object.

Feature selection is an active area in pattern recognition for two reasons. Some features introduce noise while training, which leads to mis-classification. Moreover, the additional features increase the computational complexity without much gain in classification accuracy. Therefore, we propose a feature selection method to find weights based on color separability of an object using image patch in the first frame. These weights can be determined in every frame, but the weights computed in the initial frame is sufficient for short-term videos where background do not change. The region of interest around the target is cropped from the initial frame. Since the color name features are obtained from $L*a*b$ color space, the color image is transformed into

$L * a * b$ color space. The L channel represents the lightness and two color channels, a and b represent red/green opponent colors and yellow/blue opponent colors, respectively.

Collins *et al.* (2005) used variance ratio to determine whether the foreground is separable from its surrounding background efficiently. In the proposed feature selection process, we make use of variance ratio to find whether color features are useful in separating the object from background using a and b channels of $L * a * b$ color space. A set of pixels covering the object and collection of surrounding pixels comprising the background are chosen as shown in Fig. 4.2.

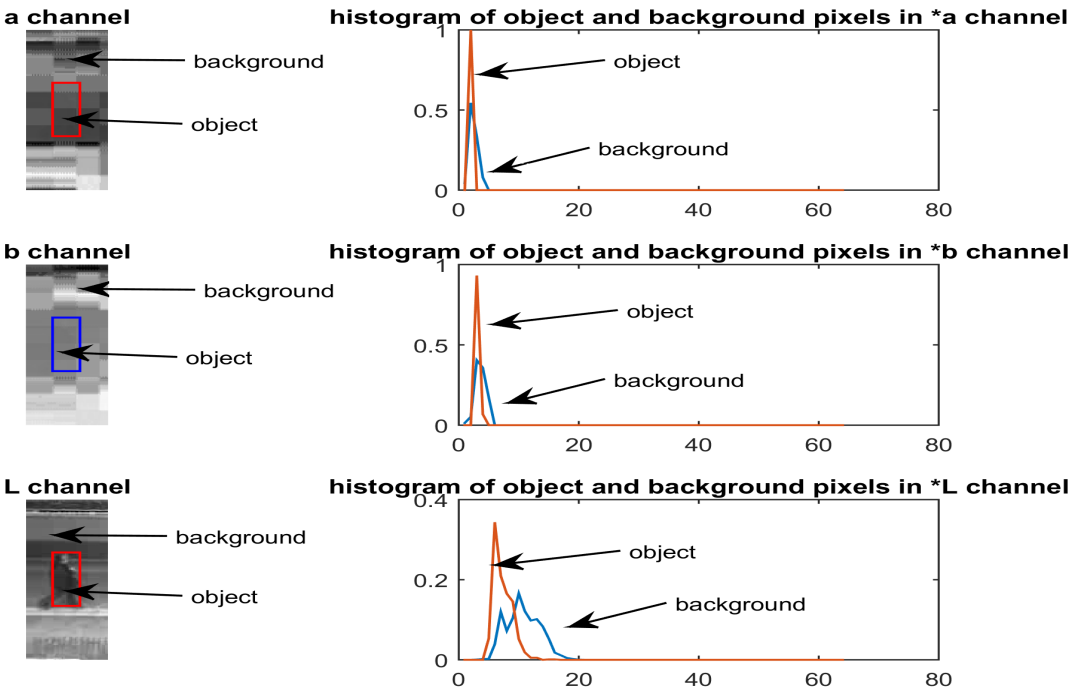


Figure 4.2: $L*a*b$ channels of image patch and corresponding object and background histograms.

Let $h_{obj}(i)$ denotes the histogram of object pixels and $h_{bg}(i)$ represents the histogram of background pixels with 64 bins. The log-likelihood ratio of pixel i is obtained from the histogram feature as follows:

$$L(i) = log\frac{max(h_{obj}(i), 0.001)}{max(h_{bg}(i), 0.001)}. \tag{4.10}$$

The nonlinear log-likelihood maps object pixels to positive value and background pixels to negative values. In order to determine the separability of object from background using $L(i)$, the variance ratio is utilized. Variance of $L(i)$ with respect to class distribution $h(i)$ is given by

$$Var(L;h) = \sum_i [h(i)L^2(i)] - (\sum_i h(i)[L(i)])^2. \tag{4.11}$$

Let $Var(L;h_{obj})$ and $Var(L;h_{bg})$ represent variance of object and background class respectively. The variance ratio is given by

$$VR = \frac{Var(L;(h_{obj}+h_{bg})/2)}{\{Var(L;h_{obj}) + Var(L;h_{bg})\}}. \tag{4.12}$$

The variance ratio indicates how well object pixels can be isolated from background pixels. Hence, it is used to find the discriminative power of color channels with respect to separability of the object from background. Let VR_a denotes variance ratio of color channel a, VR_b denotes variance ratio of color channel b and VR_L denotes variance ratio of lightness channel L. Variance ratio of the color channel is obtained by taking an average of VR_a and VR_b and is denoted as VR_{color}. Variance ratio of intensity channel is denoted as VR_L and is indicated as VR_{gray}. From observation, HoG features are more discriminative if the grayscale image is of good contrast. Hence, VR_{gray} is used for the selection of HoG features. The weights are derived based on variance ratio as follows:

$$\begin{cases} w_1 = 0.5, w_2 = 0.5, & \text{if } \mid \mathrm{VR}_{\mathrm{color}} - \mathrm{VR}_{\mathrm{gray}} \mid < 0.1 \\ w_1 = 1, w_2 = 0, & \text{else if } VR_{color} > VR_{gray} \\ w_1 = 0, w_2 = 1, & \text{else if } \mathrm{VR}_{\mathrm{color}} < \mathrm{VR}_{\mathrm{gray}} \end{cases} . \tag{4.13}$$

Thus, $w_1 = 1$ and $w_2 = 0$ select the color name feature as the most discriminative feature than HoG features, $w_1 = 0$ and $w_2 = 1$ select HoG features as the most discriminative feature than the color name feature, whereas $w_1 = 0.5$ and $w_2 = 0.5$ select both features. Thereafter, the correlation filters are trained separately using the color name features and HoG texture features under KCF framework to obtain the filter template $H1$ and $H2$ respectively as given by Eq. (4.6). Their individual responses, $\mathbf{o_{cn}}$ and $\mathbf{o_{hog}}$, for cropped image region in the present frame around the previous lo-

cation are obtained by using Eq. (4.7). The individual responses are combined using weights w_1 and w_2 to get the overall response as

$$o = w_1 \mathbf{o_{cn}} + w_2 \mathbf{o_{hog}}. \tag{4.14}$$

The position of object in the present frame is obtained as $(x_{cf}, y_{cf}) = \arg\max_{(x,y)}(o)$, where, (x_{cf}, y_{cf}) represents the location obtained through CF based tracker. In the proposed method, our objective is to find the tracking drift and switch to a modified median flow tracker by constantly observing the tracking confidence.

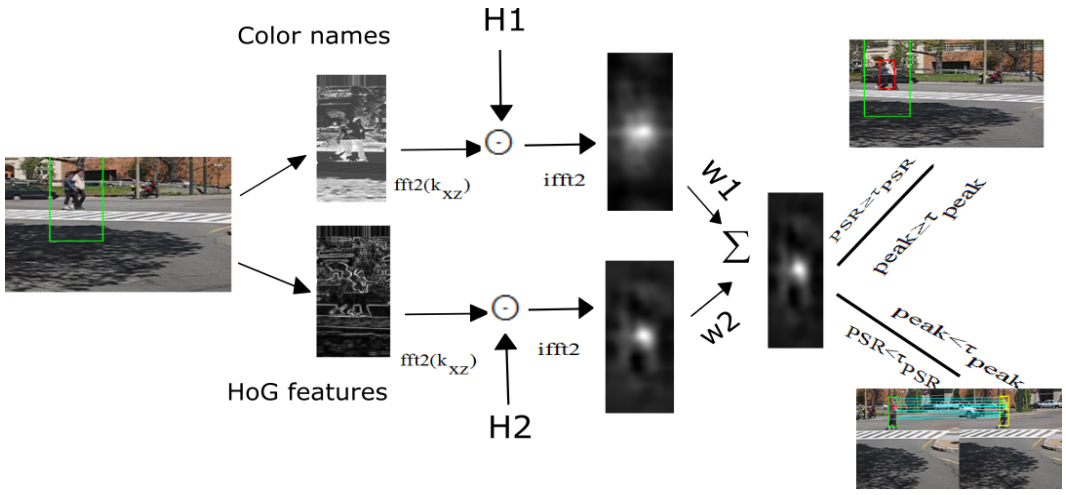


Figure 4.3: The proposed video tracking method: correlation filter based tracker switching to modified median flow tracker based on PSR and peak value of the output response.

In CF based trackers, the confidence of tracking is indicated by peak to side lobe ratio (PSR) (Bolme *et al.*, 2010). Accordingly, PSR of the correlation output response o is computed using the peak value o_p, mean μ_o and variance σ_o of the region by excluding 11×11 area around the peak value. Thus, the PSR is given by

$$PSR = \frac{o_p - \mu_o}{\sigma_o}. \tag{4.15}$$

In tracking experiments, a very low PSR indicates the possibility of occlusion or drift/tracking failure. The drifting possibilities are also observed when $peak(o) < \tau_{peak}$ and $PSR(o) < \tau_{PSR}$. The system switches to complementary tracker in order to estimate the object bounding box in the next frame, by taking bounding box of the

object in present frame. τ_{peak} and τ_{PSR} denote thresholds for peak and PSR respectively. The color/grayscale histogram of the obtained bounding box is matched with color/grayscale histogram of first ground-truth bounding box using Bhattacharyya distance metric to validate the target. The block diagram of the proposed work is depicted in Fig. 4.3.

4.2.2 Learning rate

Learning rate is used in the correlation filter based trackers to reflect the speed at which new templates are added to update the filter template. The update step is essential when new sample arrives with varying appearances. A fixed learning rate has been used in the baseline tracker. Alternately we propose to use a dynamically varying learning rate which is obtained empirically as given by

$$\eta = \frac{dist}{\sqrt{\left(1 + \left(\frac{18}{PSR+1}\right)^5\right)}}. \tag{4.16}$$

A very low value of PSR indicates the result of large appearance change, occlusion or drifting. Hence, the filter template is updated with low weights on new samples. Similarly, a high value of PSR indicates correctly classified sample; hence the filter template is updated with high weights on such samples. Variable *dist* in Eq. 4.16 specifies the Euclidean distance between previous and present locations. Moreover, *dist* indicates the moving speed of object and appearance change. Accordingly, the filter is updated based on *dist* and PSR to handle the fast moving objects, occlusion and appearance variations. The PSR values, peak values, and learning rate for all frames of *couple* sequence are displayed in Fig. 4.5.

4.2.3 Modified median flow tracker

In the proposed method, modified median flow tracker is used to recover from drifting. Even though, the median flow tracker is robust to illumination changes, it fails during fast changing illumination. In order to compensate for sudden/gradual illumination

changes, pre-processing of input frames has been performed by applying photometric normalization technique. The block diagram of photometric normalization using wavelet transform is shown in Fig. 4.4. In wavelet based normalization technique (Du *et al.*, 2005), an image is decomposed into four subbands LL, LH, HL and HH using 2D discrete wavelet transform. A normalized image is obtained by equalizing the histogram of low frequency coefficients. An edge enhancement is performed by enlarging the amplitude of high frequency components. Finally, illumination normalized image is obtained through reconstruction using inverse discrete wavelet transform.

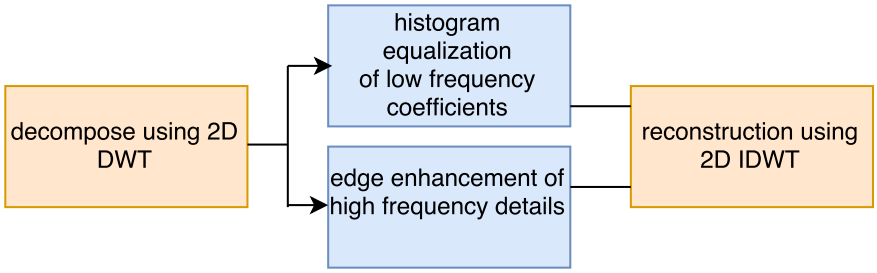


Figure 4.4: Wavelet based illumination normalization technique (Du *et al.*, 2005)

The median flow tracker is fast and accurately locates the object even when appearance changes. It may drop off tracking when there is considerable illumination change. However, it is compensated using photometric normalization technique. If the tracker misses target, then automatic tracking is hard to initialize. Hence, optical flow based tracker often requires corrections to avoid drifting. Moreover, it is unsafe to use optical flow based tracker when the bounding box in frame I_t does not contain an object. Incorrect location always results in tracking the background throughout image sequences. Since median flow tracker is not integrated with memory to learn past outputs, it entirely depends on object bounding box in the present frame. It is taken care to ensure that, the bounding box contains the valid object in current frame before switching to disjoint tracker.

To estimate the scale of object, we perform image alignment between object from the previous frame and detected object from the current frame using Lucas-Kanade method (Baker and Matthews, 2004). The thresholding is done on scale parameters to avoid abnormal increase or decrease in scale (Asha *et al.*, 2017). We make use of Piotr image and video processing toolbox (Dollár, 2009) to implement the Lucas-Kanade alignment algorithm.

4.2.4 Experimental results and discussion

4.2.5 Datasets

For experimental analysis, we consider 17 videos from OTB dataset (Wu *et al.*, 2013) with fast motion, occlusion and illumination variation as challenges. Table 2.1 provides the details of dataset used for experiments.

4.2.6 Setup

The proposed algorithm is implemented using MATLAB 15a in a machine having Intel(R) Core i5-5200U, CPU at the 2.20GHz processor with 8GB RAM. The initial location of target is obtained from the ground-truth annotation associated with the dataset. The Gaussian kernel with $\sigma_g = 0.02$ is used for mapping the input features. The regularization parameter $\lambda = 0.001$ is fixed to avoid over-fitting. The value of learning rate is computed in each frame based on PSR value and Euclidean distance between present and previous location of the object. The size of search area is fixed to twice the size of cropped image. For images, ten channel color name features and 27 channel HoG features are extracted depending on weights. The color name features and HoG features are derived from the region of interest and the weights $w1$ and $w2$ are computed according to Eq. (4.13). From experiments, we found that HoG features are more discriminative in videos like *subway*. Similarly, sequences like *bolt* is more discriminative using the color features, whereas the combination of color and HoG features perform well in sequences like *couple*. Thus, the feature selection process helps to find the discriminative features by minimizing the extraction of redundant features.

Hanning window is then applied to the search region to reduce the effect of sharp boundaries. The color and HoG feature channels of cropped image are used to train the correlation filter separately. Target by detection is implemented in the successive frames using KCF framework as described in section 4.1.1. PSR of the output response is assessed in every frame using Eq. (4.15) to decide switching. If the PSR and peak value of the output response drops below a certain threshold, tracker ignores the location obtained by CF based tracker. Thus, to find the new position, it switches

to modified median flow tracker to obtain the bounding box BB_{f+1}. In order to validate the result, the color histogram of BB_{f+1} is compared with that of initial template using Bhattacharya distance metric. The location obtained is accepted if the matching score is higher than 0.75. Once the tracker locates object, it continues to track using CF based tracker.

In the proposed work, switching to a tracker is an important stage, which switches when PSR and peak of the output response falls below a certain threshold. The threshold for PSR ranges between $0.35 \times PSR_{max}$ to $0.65 \times PSR_{max}$ and threshold for peak ranges between $0.35 \times peak_{max}$ to $0.65 \times peak_{max}$. PSR_{max} and $peak_{max}$ are considered to be PSR and peak in the second frame respectively. However, the selection of good threshold is also an important step to increase the robustness of a tracker. To make it uniform to all our experiments, we set τ_{PSR} to be $0.65 \times PSR_{max}$ and τ_{peak} to be $0.65 \times peak_{max}$ for evaluating over 17 sequences.

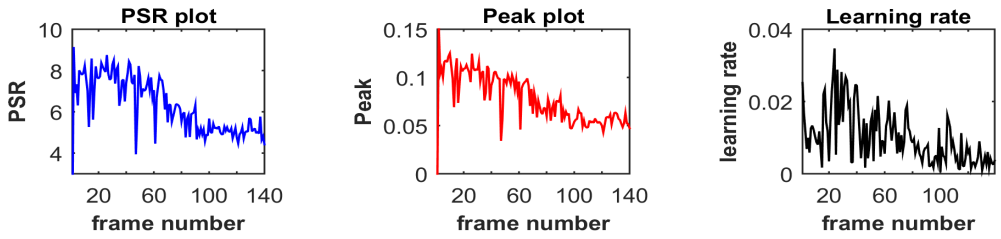


Figure 4.5: PSR (in blue) and peak (in red), learning rate (in black) for *couple* sequence. The low values of PSR plot and peak plot indicates the frame numbers where switching from CF to modified median flow tracker takes place. Learning rate plot shows the dynamic learning rate used in the proposed method

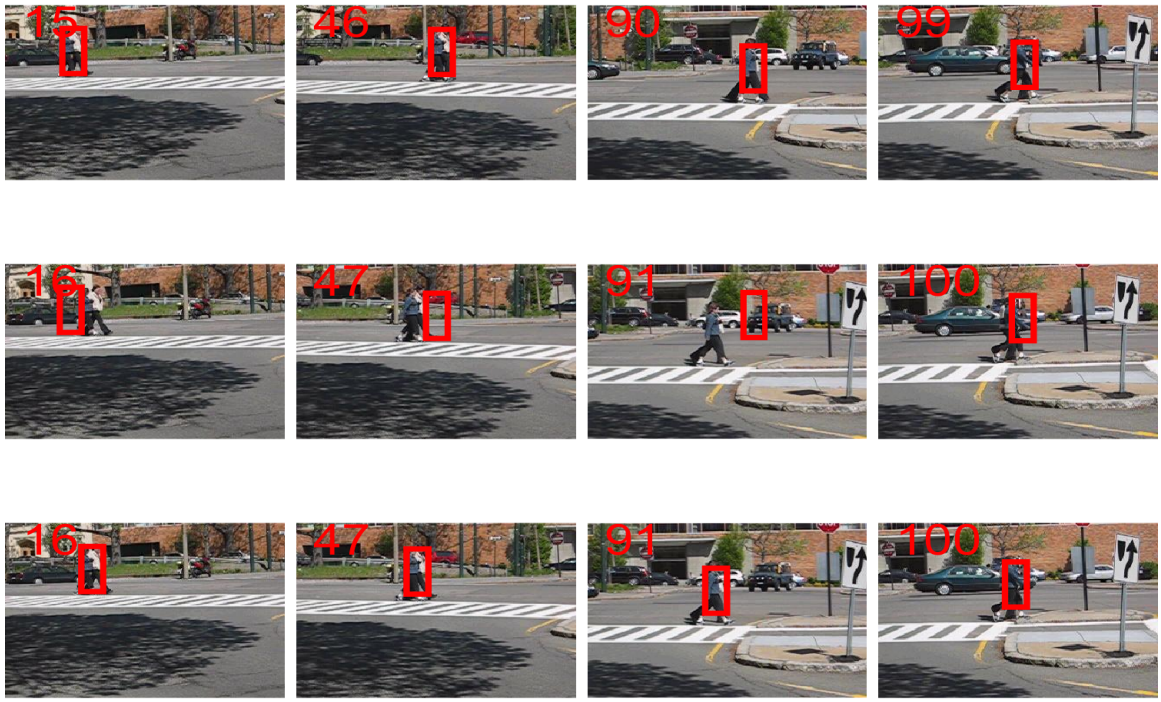


Figure 4.6: The illustration of proposed tracking method on *couple* sequence. Row1: *couple* sequence at frame number 15, 46, 90, 99. Row2: drifting illustration using CF based tracker at frame number 16, 47, 91, 100. Row3: drifting correction by switching to modified median flow tracker at frame number 16, 47, 91, 100

The proposed approach is illustrated with the help of *couple* sequence. In Fig. 4.6, the first row shows the result of CF tracker at frame number 15, 46, 90, 99. CF tracker fails to locate the object in successive frames, and is indicated by sudden drop of PSR and peak in Fig. 4.5. 2^{nd} row shows drifting of CF tracker at frame numbers 16, 47, 91, 100 where switching takes place. The tracker switches to modified median flow tracker to obtain the bounding box in the next frame i.e 16, 47, 91, 100 as shown in the 3^{rd} row.

4.2.7 Qualitative analysis

The qualitative results of state-of-the-art trackers are presented in Fig. 4.7. Occlusion is a common issue in real-time scenario, and it can be partial/full or short/long term

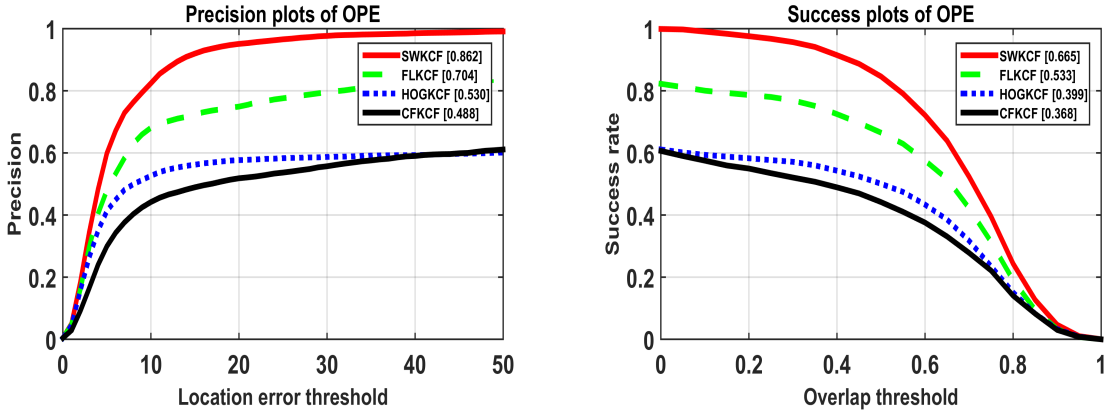
occlusions. Most of the trackers function properly when the object is partially occluded. However, these trackers tend to fail during full occlusion. The sequences like *jogging-1*, *jogging-2*, *lemming* contain partial/full occlusions. The proposed method plays well during occlusion due to varying learning rate, while KCF tracker fails to track the complete sequence. The target in *jogging-1* and *jogging-2* undergoes heavy occlusion from the pole completely, and the proposed tracker could handle efficiently. However, TLD performs well due to re-detection unit and locates the object even after occlusion. In *lemming* sequence, target disappears for short duration, trackers lose the target, and they start learning from occluded samples. However, the proposed method has a mechanism to update the filter template with low weights on occluded samples and hence successful in detecting the target when it reappears. Similarly, in *david3* sequence, the target is occluded completely by a tree two times, and TLD fails to track, while the proposed tracker tracks the person till end.

Blur is another common issue in videos that occur due to out of focus and camera or object motion. Many trackers fail to track the object in *jumping* sequence due to motion blur, whereas the proposed tracker and TLD track it completely. Illumination variation is one of the challenging cases in real-time videos, and some trackers are sensitive to light. In *shaking* sequence, the sudden variation of light is observed in many frames. The proposed tracker tracks the face, irrespective of light changes and in-plane rotations. In *couple* sequence, the object undergoes pose changes, many trackers like KCF, EDFT, DAT fail to track. However, the proposed tracker is able to follow it until the end.

Figure 4.7: Qualitative analysis of state-of-the-art trackers on challenging sequences

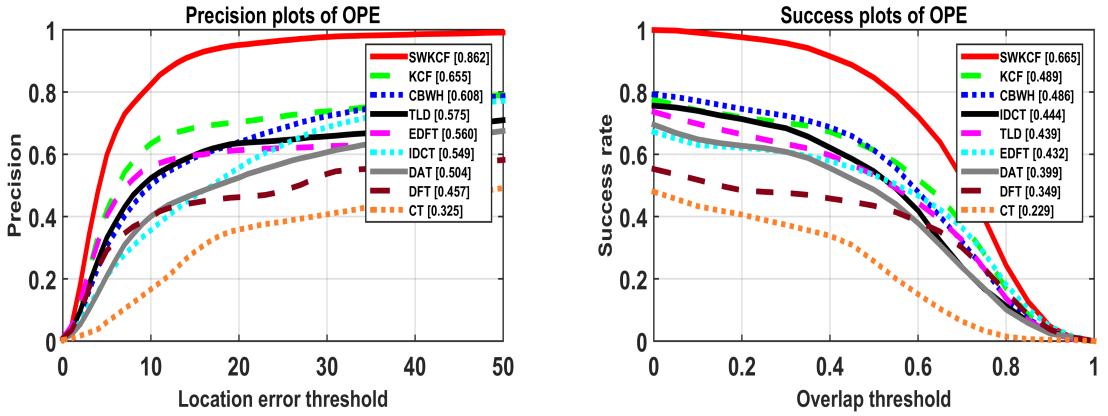
4.2.8 Quantitative analysis

The proposed method is named as SWKCF and compared with 8 benchmark baseline algorithms. The center location error (CLE) graph, distance precision score, and overlap precision score are used for quantitative analysis. We conducted one pass evaluation (OPE) test in order to evaluate the proposed method with baseline trackers. The distance precision score and overlap precision scores of trackers are presented in the Table 4.1 and Table 4.2 respectively.



(a) *Precision Plot* **(b)** *Success PLot*

Figure 4.8: Comparison of precision plot and success plots of individual properties on 17 challenging sequences of OTB dataset



(a) *Precision Plot* **(b)** *Success Plot*

Figure 4.9: Comparison of precision plot and success plots of proposed method with state-of-the-art trackers on 17 challenging sequences from OTB dataset

The proposed method outperforms other trackers on given test video sequences. To understand the strength and weakness of each technique used, we further evaluated the proposed method with only color features (CFKCF), only HOG features (HOGKCF), both color and HOG features with fixed learning rate (FLKCF), both color and HOG features with adaptive learning rate (SWKCF). The individual results are tabulated in the Table 4.3. On videos with motion blur, fast motion, background clutter, SWKCF outperforms other individual techniques. The results indicate that the proposed learning rate has a positive impact on fast motion, occlusion and motion blur. The proposed method takes advantage of HoG features that is more robust to illumination variation and deformation and color feature that is more robust for motion blur. Furthermore, learning from high PSR patches makes the system more robust. The precision plot and success plots of state-of-the-art trackers evaluated on 17 sequences are shown in Fig. 4.8.

Table 4.1: Distance precision score of state-of-the-art trackers on 17 challenging sequences

	SWKCF	**CBWH**	**CT**	**DAT**	**DFT**	**EDFT**	**IDCT**	**TLD**	**KCF**
deer	100.00	94.37	2.82	15.49	30.99	33.80	23.94	28.17	83.10
bolt	100.00	44.29	4.29	98.86	4.57	100.00	64.29	32.29	99.71
boy	100.00	100.00	66.78	100.00	48.50	100.00	95.18	100.00	100.00
jogging-1	97.72	23.45	23.13	19.54	23.45	23.13	99.02	97.39	25.08
jogging-2	100.00	100.00	16.29	20.20	34.53	16.29	16.29	96.74	16.61
doll	99.33	97.26	93.72	48.68	42.92	61.05	36.91	98.50	97.80
david3	100.00	44.44	37.30	94.84	75.00	75.00	70.63	37.30	100.00
jumping	100.00	13.42	9.90	7.67	12.78	27.80	89.14	96.81	41.53
dog1	88.74	82.30	83.56	16.74	62.37	100.00	75.93	96.44	100.00
lemming	88.70	92.37	26.27	72.68	54.42	56.89	74.55	29.72	55.76
basketball	99.59	87.31	37.38	89.93	89.52	100.00	99.72	53.24	95.86
subway	100.00	78.29	99.43	100.00	100.00	100.00	81.71	100.00	100.00
tiger1	72.78	51.00	5.73	16.62	76.50	31.52	65.04	34.67	87.97
crossing	100.00	100.00	100.00	100.00	68.33	100.00	100.00	60.00	100.00
couple	100.00	67.14	40.71	66.43	10.71	11.43	54.29	32.14	27.86
shaking	96.44	2.74	4.11	3.29	83.01	16.71	1.37	43.29	4.11
surfer	98.40	85.64	0.80	100.00	3.72	100.00	23.67	99.73	92.02

Table 4.2: Overlap precision score of the state-of-the-art trackers on 17 challenging sequences

	SWKCF	**CBWH**	**CT**	**DAT**	**DFT**	**EDFT**	**IDCT**	**TLD**	**KCF**
deer	100.00	90.14	2.82	9.86	30.99	33.80	23.94	28.17	81.69
bolt	90.00	41.71	0.57	96.00	4.00	99.43	54.86	17.71	94.29
boy	99.17	98.84	59.97	96.35	48.34	98.34	89.87	84.72	99.17
jogging-1	96.74	22.48	21.50	18.89	21.50	22.15	96.09	96.42	22.48
jogging-2	100.00	97.72	2.61	19.54	15.64	15.31	14.98	95.44	15.96
doll	68.85	72.13	56.46	18.31	35.02	49.35	13.43	42.79	55.22
david3	95.63	93.65	26.98	100.00	74.21	71.43	96.43	31.75	99.21
jumping	99.36	10.22	0.96	5.43	11.82	14.38	53.67	92.33	28.12
dog1	59.85	62.96	62.30	6.52	52.15	65.04	49.48	72.00	65.11
lemming	83.31	88.17	25.45	74.40	47.38	48.20	69.99	21.71	44.24
basketball	97.93	86.21	5.93	89.52	71.59	89.93	99.03	35.86	89.79
subway	100.00	64.57	47.43	90.86	99.43	99.43	79.43	97.14	100.00
tiger1	64.18	44.99	1.72	14.33	67.91	29.23	56.73	32.95	85.67
crossing	94.17	82.50	87.50	97.50	64.17	98.33	82.50	45.83	95.00
couple	67.14	38.57	30.71	63.57	8.57	10.71	34.29	22.86	24.29
shaking	86.30	1.64	3.01	3.01	82.47	16.16	1.10	39.45	1.37
surfer	35.90	46.01	0.27	22.34	3.72	45.48	11.17	85.64	39.89

Table 4.3: Distance precision and overlap precision scores of individual methods of the proposed method

	Distance Precision				Overlap Precision			
	CFKCF	**HOGKCF**	**FLKCF**	**SWKCF**	**CFKCF**	**HOGKCF**	**FLKCF**	**SWKCF**
deer	15.49	2.81	**100**	**100**	12.67	2.81	**100**	**100**
bolt	31.71	2.28	**100**	**100**	28.57	1.14	**98.85**	90
boy	97.67	**100**	**100**	**100**	95.68	**99.16**	**99.16**	**99.16**
jogging-1	97.06	23.45	24.42	**97.71**	89.57	22.47	22.47	**96.74**
jogging-2	89.57	16.28	16.61	**100**	88.59	14.98	15.30	**100**
doll	**99.32**	98.26	99.17	**99.32**	68.85	**69.49**	69.13	68.85
david3	17.85	**100**	**100**	**100**	15.47	93.65	92.06	**95.63**
jumping	69.00	**100**	**100**	**100**	37.69	99.36	99.68	**99.36**
dog1	86.37	88.74	**100**	88.74	**62.88**	59.85	63.18	59.85
lemming	4.11	39.44	45.35	**88.69**	3.89	38.99	40.71	**83.30**
basketball	**99.58**	4.27	31.86	**99.58**	97.51	2.06	28	**97.93**
subway	24	**100**	**100**	**100**	18.28	**100**	**100**	**100**
tiger1	72.77	**98.28**	72.77	72.77	64.18	**97.13**	64.18	64.18
crossing	57.5	**100**	65	**100**	35.83	**94.16**	52.5	**94.16**
couple	10.71	11.42	65	**100**	7.85	10	57.14	**67.14**
shaking	7.39	6.02	**98.90**	96.43	5.75	4.10	**89.31**	86.30
surfer	32.18	**100**	**100**	98.40	17.55	**41.48**	39.09	35.90

4.2.9 Timing complexity

The overall complexity of the proposed tracker is based on that of KCF tracker and modified median flow tracker. The timing complexity of KCF tracker is decided upon area of region selected and is given by $O(nlogn)$. In this, n denotes the number of pixels present in the area. Similarly, timing complexity of LK tracker is $O(n^2N + n^3)$, where n is 2 for 2D optical flow tracking and N is the number of pixels in a template. The timing complexity of the proposed tracker is same as KCF tracker with extra overhead due to switching to median flow tracker.

4.3 Summary

In this chapter, an improved version of kernelized correlation filter based tracker has been proposed by switching to modified median flow tracker to achieve drift free tracking. The proposed method learns discriminative correlation filter using weighted color and HoG feature channels. The weights have been derived based on color separability of foreground from background of a patch in a particular video. It also incorporates Lucas Kanade method to estimate the scale parameters effectively. We further proposed an adaptive learning rate to adjust with changing appearance of an object and occlusion, which is computed using PSR of the output response. Overall, the proposed method showed comparatively better performance on challenging sequences against most recent state-of-the-art baseline trackers. In addition, the switching technique is faster, hence it can be used for real-time application. However, selection of number of trackers, the order of evaluation and optimum value of threshold for switching need further investigation.

Chapter 5

INFRARED TARGET TRACKING

Designing an efficient tracker to obtain the trajectory of an object in thermal infrared video is a tedious task due to textureless and colorless properties of an object. In this chapter, a combination of discriminative and generative approaches is realized to improve the accuracy of a tracker. In discriminative approach, kernelized correlation filter with spatial features and AdaBoost classifier with intensity features are combined. The object locations are gathered using above approaches by running them in parallel. Further, these locations are fine-tuned using the generative technique to acquire best target location based on linear search method. Finally, the scale of target is determined by applying Lucas-Kanade homography estimation algorithm. The proposed technique is tested using 17 challenging infrared videos selected from LTIR dataset. Besides, quantitative and qualitative assessment of the proposed approach is compared with the state-of-the-art trackers. Thus, section 5.1 presents the proposed approach for tracking, section 5.2 provides details of experimental analysis followed by summary in section 5.3.

5.1 Combined approach for tracking

Track-by-detect approach considers tracking as a classification task to obtain the position of target in every frame. In general, peak classification score corresponds to the best location of target, however, this hypothesis sometimes makes the tracker to predict wrong location. Hence, two approaches are employed in parallel based on complementary features, i.e, spatial structure and pixel intensity values to determine the object location. The main steps of the proposed method are as follows:

Step 1: Initially, gradient and channel coded feature maps of the patch are derived to train KCF filters separately. The trained filter is then correlated with corresponding features of the region around present location. The output responses obtained from individual filter are fused adaptively to locate the target based on peak value. Similarly, a patch comprising object pixels and background pixels are trained using AdaBoost classifier to segment the object in every frame. Later, the mean-shift algorithm is applied to the segmented object region to determine the position of target. This step is depicted in Fig. 5.1 and Fig. 5.2.

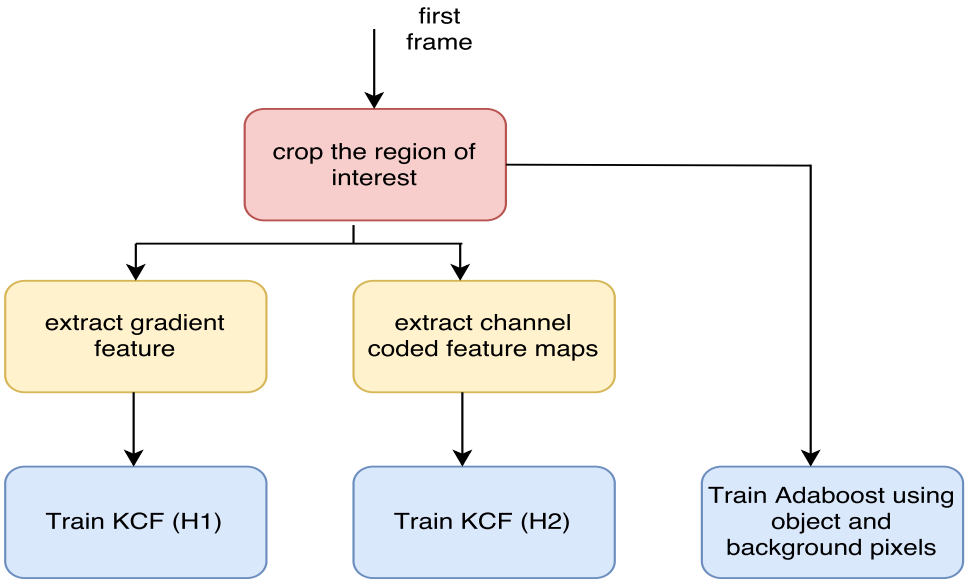


Figure 5.1: Block diagram of training phase using KCF and AdaBoost classifier

Step 2: The object locations retrieved in the step 1 denote possible positions of the object. The best location is determined based on maximum NCC score between

the area around that location and object model. The object model, filter template, and classifier model are updated in every frame based on confidence score.

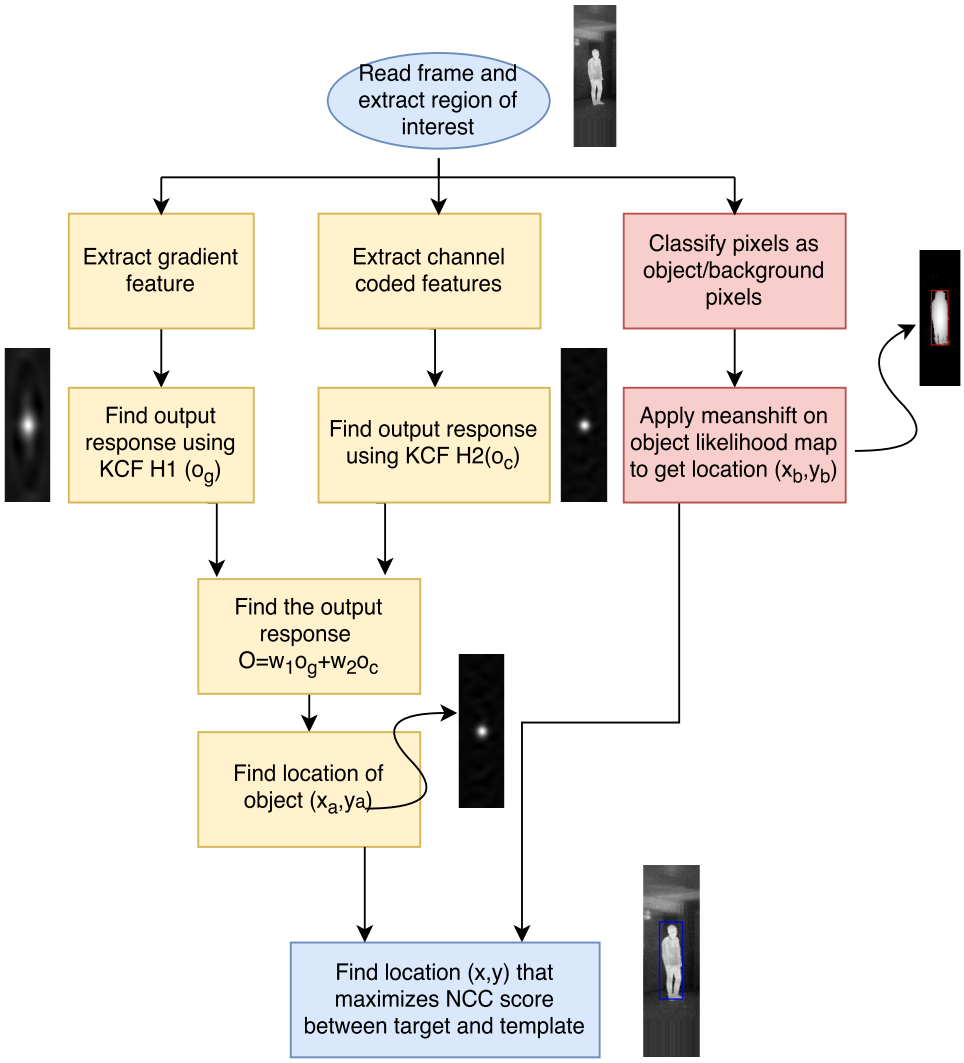


Figure 5.2: Localization of object using kernelized correlation filter and AdaBoost classifier

Step 3: The scale of object is determined in every frame based on Lucas-Kanade homography estimation method.

The above steps are repeated until the last frame. The following subsections provide further details of the proposed approach.

5.1.1 KCF tracker

Correlation filter based tracker considers tracking as a problem related to template matching through convolution operation to produce sharp peak for the desired target. Thus, the greatest value of output probability map indicates the position of object. KCF tracker using a single feature has been reviewed in the section 4.1.1. In the proposed approach, KCF is extended utilizing two set of features separately and is discussed in the following subsection.

5.1.1.1 Feature sets

Selection of best features is important in tracking experiments as thermal images lack texture and color attributes. In this work, the gradient feature furnish edge details while channel coded feature maps provide the details of intensity distributions to describe an object. The gradient feature is observed to be robust for temperature and contrast changes. Hence, the integral gradient is used as one of the feature channels as shown in Fig. 5.3(a). The edge image $I_g = \sqrt{I_x^2 + I_y^2}$ is formed by computing gradients I_x and I_y in x and y orientations respectively.

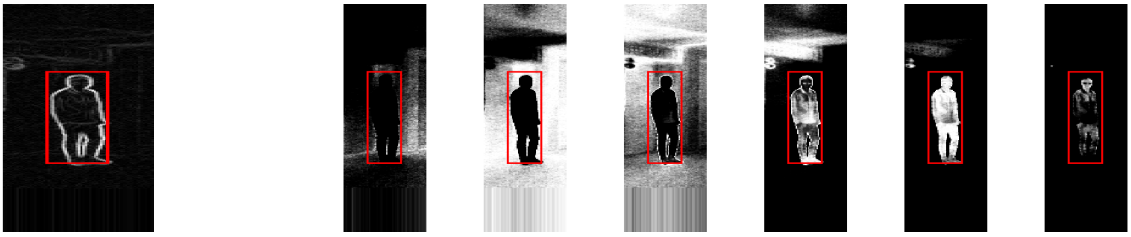
The channel representations (CR) are biologically inspired data representations which is widely applied in machine vision and tracking applications (Felsberg, 2013), (Jonsson, 2008). Fig. 5.3(b) depicts the channel coded features (Jonsson, 2008). A channel vector c is constructed from scalar y using a nonlinear transformation and is obtained as

$$\mathbf{c} = [K(y - y_1), K(y - y_2).....K(y - y_n)]^T, \tag{5.1}$$

where $K\{.\}$ represents a symmetric non negative basis function with y_1, y_2,..y_n as channel centers or bin centers. Thus, CRs are generated from a scalar using kernel functions $K\{.\}$ such as cos^2 for binning the data to get a smooth histogram. In this, n samples of data x_i with each sample representing a pixel value of an image is encoded. Finally, the coefficients of CRs are obtained from data x_i and bin centers r with a spacing of h as:

$$\mathbf{c_r} = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x_i}{h} - r\right). \tag{5.2}$$

The spacing parameter $h = 22$ is chosen for better representations of data (Felsberg, 2013). The dynamic range of pixels in infrared images is small when compared to RGB images, hence the number of channels in CRs are obtained as $\left[\frac{x_{\max}-x_{\min}}{h}\right] + 2$ and encoding interval is chosen as $I = [x_{\min}\ x_{\max}]$, where x_{max} and x_{min} represents the maximum and minimum value of x respectively. The basis function $K\{.\}$ is chosen as cos^2 to generate the soft histogram with overlapping bins and samples are weighted relative to the distance from center.



(a) *gradient* **(b)** *channel coded feature maps*

Figure 5.3: Gradient and channel coded feature maps used in the proposed method

5.1.1.2 Multi-feature KCF tracker

The need for working with multi-features rather than single feature is to distinguish both shape and intensity features efficiently. Therefore, the gradient and channel coded intensity features are employed in the work. Each channel is multiplied by cosine window to eliminate the effects of sharp boundaries due to circular convolution.

Initially, KCF is trained separately using gradient and channel coded feature maps to obtain filters H_1 and H_2 respectively as depicted in Fig. 5.1. The gradient feature promotes edge information to build a stable appearance model, while channel coded intensity maps give the details of intensity distribution. The filter responses corresponding to the gradient (o_g) and channel coded features (o_c) are combined in every frame, as shown in Fig. 5.2. The adaptive weights are computed based on PSR to combine the individual output response maps. As mentioned earlier, PSR is a measure used to decide the strength of peak in output response map, where each pixel in confidence map indicates the probability of a pixel location belonging to the object.

The weights are computed as follows:

$$w_1 = \frac{PSR_g}{PSR_c + PSR_g} \tag{5.3}$$

and

$$w_2 = \frac{PSR_c}{PSR_c + PSR_g}, \tag{5.4}$$

where PSR_g represents PSR of gradient response and PSR_c denotes PSR of channel coded feature response.

The confidence maps are combined to generate the fused map o using weights w_1 and w_2 as $o = w_1 o_g + w_2 o_c$. The advantage of using adaptive weights is that the gradient feature performs superior to channel coded features in some sequences and a large weight is assigned to gradient feature than channel coded features to improve the tracking performance. Similarly, large weight is assigned to channel coded features for some sequences where intensity distribution feature dominates edge features. Finally, the location of object is obtained corresponding to the maximum value of output response o and is denoted as (x_a, y_a) or l_{cf}.

5.1.1.3 Template update step

The filter needs to be updated in every frame to accommodate recent object appearances. Baseline tracker (KCF) uses fixed learning rate to update the filter template in every frame as given by Eq. (4.5) and Eq. (4.6) to control the speed of tracker. However, the correlation filter is very sensitive to deformation, occlusion and large appearance changes. The probability of drift increases when the filter template is updated with occluded samples. In the proposed work, the filter template is updated depending on PSR value. Consequently, Eq. (4.5) and Eq. (4.6) are used with the normal value when PSR of output response is above a pre-defined threshold. During substantial appearance changes and occlusion, the learning rate is reduced to a small value. The above-mentioned procedure minimizes the drift to a large extent.

5.1.2 Pixel based target segmentation

The correlation filter model learns using the spatial structure which is sensitive to deformation and occlusion. In contrary, model based on pixel classification is robust to shape changes but sensitive to contrast. To avail the advantages of these approaches, this work suggests combining the two techniques running in parallel to increase the accuracy. AdaBoost classifier is used in literature to distinguish object pixels from surrounding pixels (Avidan, 2006) for color images. In the proposed method, we used AdaBoost classifier to classify the object pixels from the background pixels in infrared imagery. The region of interest with 1.5 times size of the object is cropped from current location. The training samples include a set of 8×8 patches around the object and background pixels to serve as positive and negative examples respectively.

Algorithm 2 Adaboost classifier for foreground segmentation

input: Object and background region
output: Object likelihood map.

Training Stage
generate Object and background patches of size 8x8 around each pixel. i.e. $\{x_i, y_i\}\big|_{i=1}^{N}$, $y_i = 1$ for foreground and $y_i = 2$ for background.
Set the weight for each sample i.e. $w_1(i) = \frac{1}{N}$
for t = 1 to T **do**
Apply decision stump D_t to classify each sample
Compute the error (number of misclassification over target set size) ε_t
Determine the weak classifier weight $\alpha_t = \frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
Update the sample weights $\mathrm{w}_{t+1}(i) = \frac{w_t(i)\exp(-\alpha_t y_i D_t(x_i))}{\sum_{i=1}^{N} w_t(i)}$
end for
Combine weak classifiers to get strong classifier as $D = sign\left(\sum_{t=1}^{T} \alpha_t D_t\right)$
Testing Stage
Generate patches of size 8x8 around each pixel. $\{x_i, y_i\}\big|_{i=1}^{N}$
Categorize each patch as foreground or background using the classifier D

The trained model is applied to classify unlabeled pixels as outlined in Algorithm 2. The model updation is done in every frame based on tracking confidence. Two measures are utilized; they include PSR and aggregate of weight map. Aggregate (sum) of weight map is the algebraic addition of pixel values in the likelihood map

obtained from the classifier, likely to reduce during occlusion and increase due to background clutter. If both measures are above a pre-defined threshold, the classifier is updated with the present target region.

The location of target in the present frame acts as starting position for the mean-shift algorithm. The location of object in the next frame is obtained using the shift of centroid in the object likelihood map. In every iteration, the new position of object is moved to the centroid of object till mean-shift converges. The center of an object located using mean shift is expressed as (x_b, y_b) or l_{ab}.

5.1.3 Template matching using NCC

The generative method is utilized to get the actual target location from the positions obtained through discriminative techniques. Two locations namely (x_a, y_a) and (x_b, y_b) are obtained by correlation filter and AdaBoost classifier based approaches respectively. The generative technique such as normalized cross correlation is applied on probable target patches to select the best target. The l_{cf} or (x_a, y_a) denotes the location obtained by KCF tracker and l_{ab} or (x_b, y_b) represents the location obtained by AdaBoost classifier approach. The probable object position is given by

$$l_f = w_l l_{cf} + (1 - w_l) l_{ab}, \tag{5.5}$$

where weight w_l takes values from 0 to 1. A patch of size $P \times Q$ is cropped from the location l_f and is denoted as T_{lf}. The template matching is accomplished in two steps. The initial step is carried out by comparing the probable target with previously obtained target t_{n-1}. Thus, similarity between probable target T_{lf} and previously detected target t_{n-1} is represented as $S_{(T_{lf}, t_{n-1})}$ using NCC score. Thereafter, the probable target is compared with the multi-frame template. Similarly, similarity between T_{lf} and multi-frame template T_{mt-1} using NCC score is denoted as $S_{(T_{lf}, T_{mt-1})}$. The multi-frame template is constructed by computing the running average of targets detected in the previous frames as given by Eq. (5.6). The multi-frame template is calculated as

$$T_{mt} = \lambda_t T_{mt-1} + (1 - \lambda_t) t_{n-1}. \tag{5.6}$$

The multi-frame template is updated with the learning rate of $\lambda_t = 0.05$ to include recent appearances. The weighted combination of $S_{(T_{lf},t_{n-1})}$ and $S_{(T_{lf},T_{T_{mt-1}})}$ is maximized with respect to w_l. The optimum weight w_l is used to decide the final location of target i.e.,

$$\arg\max_{w_l}(\theta S_{(T_{lf},t_{n-1})} + (1-\theta)S_{(T_{lf},T_{T_{mt-1}})}). \tag{5.7}$$

NCC template matching has been extensively used in tracking and detection literature for several years. NCC score between 2 vectors $\mathbf{x}$ and $\mathbf{y}$ is pixel by pixel comparison and mean value is subtracted to eliminate the effect of intensity variation. The correlation value is normalized to produce the score in the range 0 to 1. 0 represents a strong mismatch, while 1 denotes a strong match between two vectors $\mathbf{x}$ and $\mathbf{y}$. The NCC (Briechle and Hanebeck, 2001) score is given by

$$NCC_{x,y} = \frac{\sum_{i=1}^{N}\left((x_i-\mu_x)\left(y_i-\mu_y\right)\right)}{\sqrt{\sum_{i=1}^{N}\left(x_i-\mu_x\right)^2\sum_{i=1}^{N}\left(y_i-\mu_y\right)^2}}. \tag{5.8}$$

5.1.4 Scale estimation

Lucas Kanade method is very popular to estimate the affine parameters of warped image Baker and Matthews (2004). In the proposed scale estimation problem, we consider matching of template in the previous frame and current frame to find the scale parameters. The objective of Lucas Kanade method is to generate the set of parameters that minimizes the sum of squared error between reference image T (target template obtained in the previous frames) and the warped image I (target template obtained in the current frame). In general, minimization problem is treated as follows:

$$\sum_{\mathbf{x}}\left(I\left(\mathbf{W}(\mathbf{x};\mathbf{p})\right) - T(\mathbf{x})\right)^2 \tag{5.9}$$

where $\mathbf{W}(\mathbf{x};\mathbf{p})$ denotes set the warps based on parameters $\mathbf{p} = (p_1, p_2, p_3, p_4, p_5, p_6)^T$. Thus, $\mathbf{W}(\mathbf{x};\mathbf{p})$ takes the pixel $\mathbf{x}$ in the coordinate frame of the template and maps it to the sub-pixel location $\mathbf{W}(\mathbf{x};\mathbf{p})$ in the coordinate frame of the image I. The sub-pixel

locations are computed as

$$\mathbf{W}(\mathbf{x};\mathbf{p}) = \begin{pmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{5.10}$$

In the proposed approach, we considered scale parameters only. Hence, $\mathbf{W}(\mathbf{x};\mathbf{p})$ changes to,

$$\mathbf{W}(\mathbf{x};\mathbf{sc}) = \begin{pmatrix} sc(1) & 0 & 0 \\ 0 & sc(2) & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \tag{5.11}$$

Lucas Kanade algorithm assumes the value of $\mathbf{p}$ and iteratively solves for increments to obtain the parameters $\mathbf{\Delta p}$. Hence, the following expression is minimized with respect to $\mathbf{\Delta p}$.

$$\sum_{\mathbf{x}} \left(I\left(\mathbf{W}(\mathbf{x};\mathbf{p}+\mathbf{\Delta p})\right) - T(\mathbf{x})\right)^2 \tag{5.12}$$

Thereafter, $\mathbf{p}$ is changed to

$$\mathbf{p} \leftarrow \mathbf{p} + \mathbf{\Delta p} \tag{5.13}$$

The above two steps are iterated until $\mathbf{p}$ converges.

Algorithm 3 presents the proposed scale estimation algorithm based on Lucas-Kanade homography estimation method. The objective of Lucas-Kanade technique is to minimize sum of squared difference (SSD) error between two patches. An image patch is warped using scale parameters [sc(1) sc(2)]. The optimum scale factor is computed through an iterative process using gradient descent algorithm. Lucas-Kanade algorithm assumes initial scale parameters and solves for the best value till convergence is attained. In the proposed method, object template of previous frame and target obtained in the current frame are employed to estimate the scale parameters. The scale limiting thresholds are considered to avoid abnormal increase or decrease in scale. The target size and position are updated based on scale factor obtained.

Algorithm 3 Proposed scale estimation algorithm

input: current target_size, target_pos and current_scale_factor, object region at frame number $t-1$ and t

output: updated target_size, target_pos and current_scale_factor.

perform image alignment between object region at frame number $t-1$ and object region at frame number t using Lucas-Kanade method (Baker and Matthews, 2004) to estimate scale parameters [sc(1) sc(2)]
if $sc(1) \geq t_{up}$ **then** $sc(1) \leftarrow t_{up}$ **end if**
if $sc(1) \leq t_{dn}$ **then** $sc(1) \leftarrow t_{dn}$ **end if**
if $sc(2) \geq t_{up}$ **then** $sc(2) \leftarrow t_{dn}$ **end if**
if $sc(2) \leq t_{dn}$ **then** $sc(2) \leftarrow t_{dn}$ **end if**
current_scale_factor=$\frac{2}{sc(1)+sc(2)}$
target_size = target_size*current_scale_factor
target_pos = target_pos*current_scale_factor

5.2 Experimental analysis

5.2.1 Setup

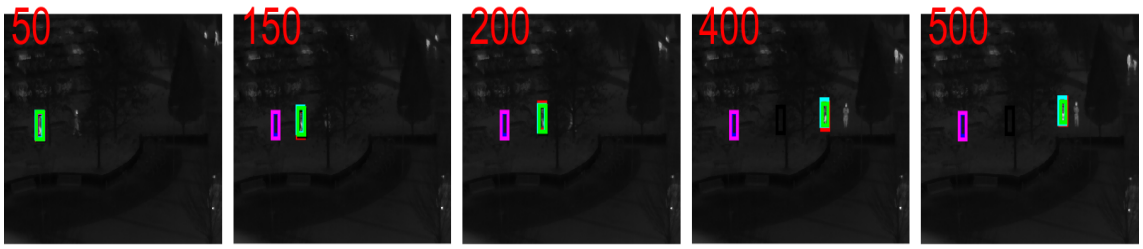
The proposed algorithm is implemented using MATLAB 15a software in a machine with intel(R) core i5-5200U, CPU with 2.20GHz processor and 8GB RAM. The proposed approach is evaluated using 17 challenging sequences from LTIR dataset as listed in Table 2.2. The challenges include pose change, occlusion, scale variation, temperature changes and so on. The ground-truth annotation provides the location of object in the first frame to start tracking. The proposed method comprises 3 approaches. First approach involves KCF tracker using gradient and channel coded features. The parameters used in the KCF tracker are as follows: The size of patch is 1.5 times that of the object. The feature channels are multiplied by cosine window to smooth the boundaries. The input features map to Gaussian kernel space with $\sigma_g = 0.02$. The regularization parameter $\lambda = 0.001$ is set to avoid over-fitting. The learning rate is kept at 0.025 for high confident frames and reduced to 0.001 for low confident frames. Two confidence measures include PSR and aggregate of object likelihood map. In the proposed experiments, a threshold for PSR is chosen to be $0.5 \times PSR_{max}$ and threshold for aggregate of weight map is selected as $S \leq 2 \times W_p$ & $S \geq 0.5 \times W_p$

to update the filter template, where W_p indicates the sum of weight map of starting frame.

Further, AdaBoost classifier is employed to classify the pixels into object and background classes. The location of object is detected by performing the mean-shift operation on likelihood map. Ultimately, NCC based template matching refines the object locations retrieved using discriminative approaches. To find probable locations, weight w_l is varied from 0 to 1 in steps of 0.5. The object region is cropped and compared with previous template t_{n-1}, also with multi-frame template T_{mt-1}. To incorporate changes in object appearances, multi-frame template T_{mt} is calculated as running average of targets obtained from starting frame with the learning rate of $\lambda_t = 0.05$. The weight w_l and corresponding location of an object is attained by maximizing the weighted combination of NCC scores. The weight θ is fixed to give more importance to the multi-frame template than previously detected target. In the proposed method, θ is fixed at 0.1 throughout the experiments. The scale parameters are estimated using Lucas-Kanade scale homography estimation. The scale limiting thresholds are used to avoid unexpected increase/decrease in the scale. The $t_{up} = 1.02$ and $t_{dn} = 0.98$ are used to maintain the stability. Piotr image and video processing toolbox (Dollár, 2009) has been used to realize the fast version of AdaBoost classifier and Lucas-Kanade algorithm.

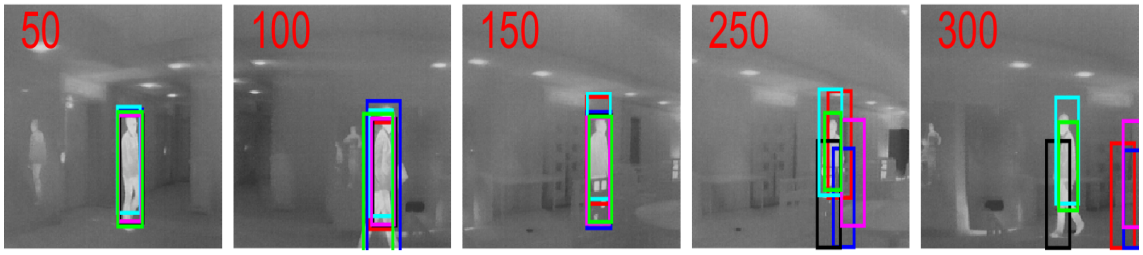
5.2.2 Qualitative analysis

In this section, the proposed tracking method is evaluated with five baseline state-of-the-art trackers. The five trackers FCT (Zhang *et al.*, 2014), EDFT (Felsberg, 2013), DSST (Danelljan *et al.*, 2014), KCF (Henriques *et al.*, 2015), and DAT (Possegger *et al.*, 2015) are used for both visual and thermal infrared target tracking. For comparison purpose, source code provided by authors is used with tuned parameters as mentioned in the respective papers.

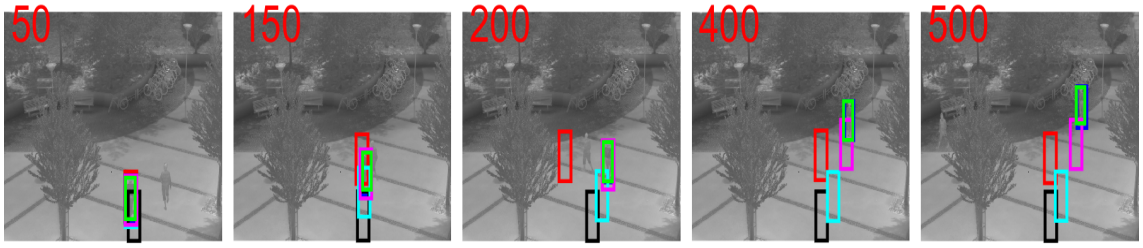


(a) *trees1*

(b) *crouching*



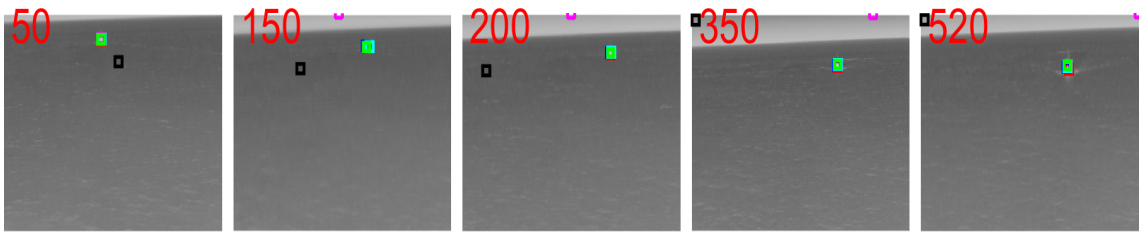
(c) *hiding*



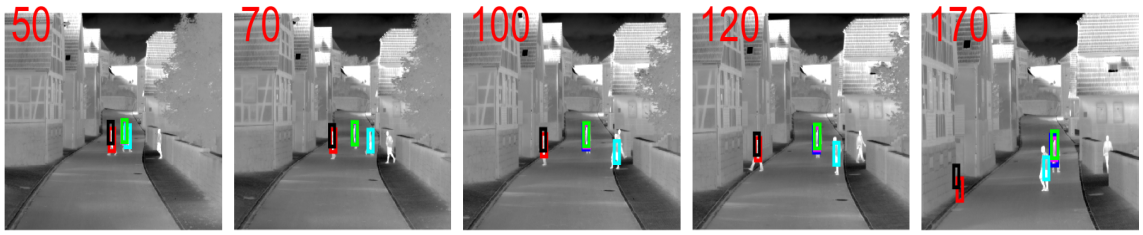
(d) *depthwise crossing*

(e)

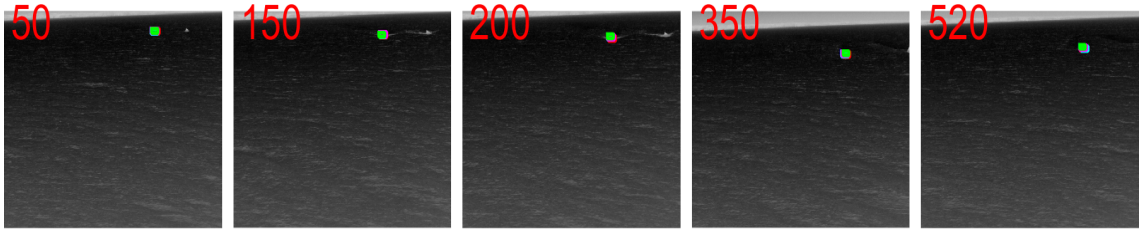
Figure 5.4: Tracking of a person in *trees1*, *crouching*, *hiding*, *depthwise crossing* image sequences.



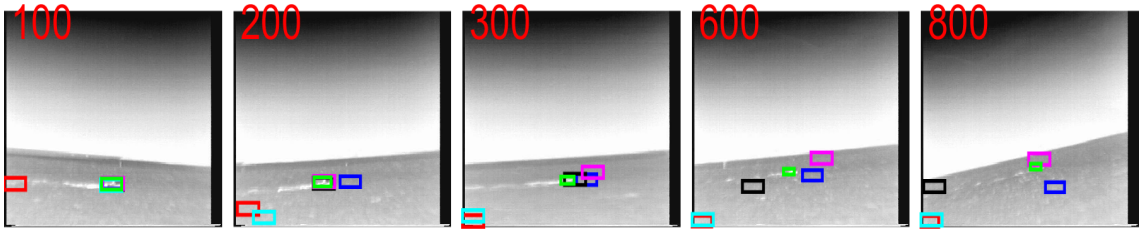
(a) *boat1*



(b) *street*



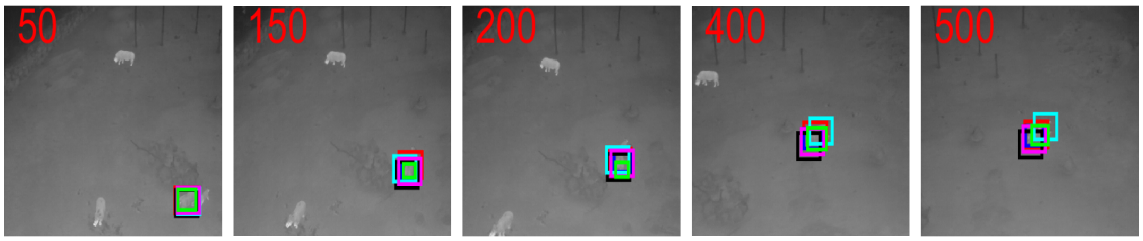
(c) *boat2*



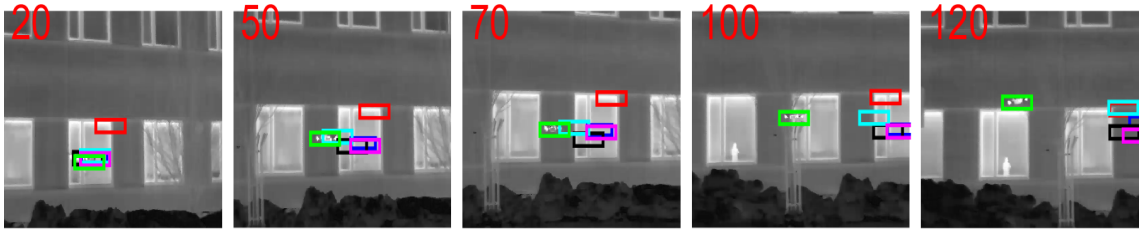
(d) *ragged*

(e)

Figure 5.5: Tracking of a boat in *boat1*, *street*, *boat2*, *ragged* image sequences.

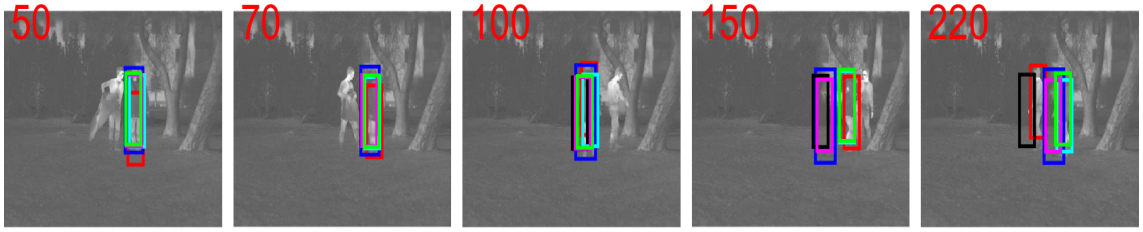


(a) *running rhino*



(b) *quadrocopter*

(c) *jacket*



(d) *birds*

(e)

Figure 5.6: Tracking of a rhino in *running rhino*, a quadrocopter in *quadrocopter*, a person in *jacket*, *birds* image sequences.

A set of experiments have been carried to analyze the proposed method, and sample frames are illustrated in Fig. 5.4, Fig. 5.5, and Fig. 5.6. In *trees1* image sequence

as shown in Fig. 5.4(a), the person walks in an area surrounded by trees gets occluded several times behind the trees. Besides, there are many people walking in the same scene. The objects appear as a bright spot (at a higher temperature than the surroundings) when compared with background. The proposed tracker, DAT, and FCT successfully track the person irrespective of occlusion and background clutter. In Fig. 5.4(b), *crouching* sequence has less resolution and poor contrast due to comparable pixel intensities of the object and background. Further, the person appears in walking and sitting poses. In addition, the object is occluded in the frames 333 up to 579 by a similar object. The proposed tracker achieves good results as compared to other trackers. In Fig. 5.4(c), *hiding* sequence portrays the indoor environment with a person moving around and hiding behind another bigger object completely. The full occlusion happens at frame number 135 to 288. Moreover, the object experiences substantial scale changes due to moving camera, which makes tracking task more challenging. DAT, DSST, and KCF lose target due to occlusion and also fail to re-detect the object when it re-enters. The possible reason may be learning from inaccurate samples due to total occlusion. FCT and EDFT can re-detect the object when it re enters but fails to locate it properly. The proposed method is the only tracker to follow the object with proper scale estimation and able to re-detect after occlusion. But the proposed method fails to estimate the scale accurately after occlusion. In *depth-wise crossing* sequence depicted in Fig. 5.4(d), the target is at a lower temperature than the surroundings, hence appears darker compared to background. The proposed tracker and DSST outperform other investigated trackers.

The target in *boat1* sequence as illustrated in Fig. 5.5(a) contains a boat as the moving object over the surface of water. The images are captured using a fixed camera, but the target has changing appearance and scale. The *street* sequence shown in Fig. 5.5(b), depicts group of people walking in the street. Tracking a person is very challenging due to constant movements, sharing similar shapes and intensity values. Furthermore, the object gets occluded by another object of similar class in frame numbers ranging between 6 to 54 and 146 to 165. Trackers like FCT, EDFT easily lose or shift track as they search in the local area but work well when target appearances change slowly. The proposed method and DSST show least drifting error and are very effective in localizing the target in every frame. In *boat2* sequence shown in Fig. 5.5(c), initially the target *boat* is very small, but gradually enlarges as it moves close to the proximity of camera. The proposed tracker can track till the end of sequence

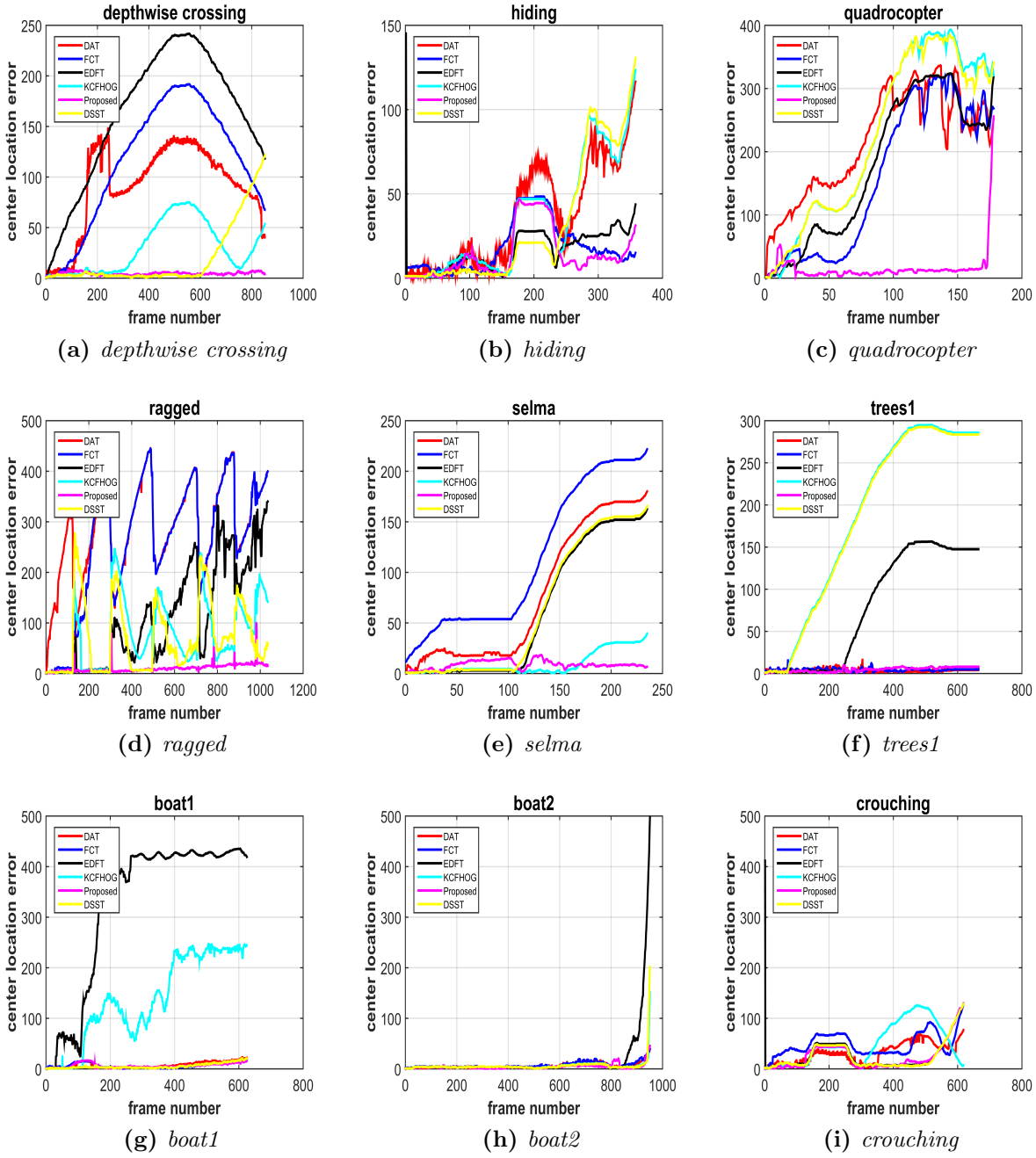
irrespective of change in the object size and appearance. The *ragged* in Fig. 5.5(d) is another video containing a boat as target in which the object moves swiftly with disappearance and appearance from camera view several times. The state-of-the-art trackers like DAT, DSST, EDFT, FCT fail to track during first occlusion itself and cannot re-detect the object. Only the proposed tracker is able to monitor the sequence completely while effectively handling scale variation.

The *running rhino* sequence in Fig. 5.6(a), contains a *rhinoceros* as a moving object in an area containing many trees and other animals belonging to the same class. Although the sequence has many similar objects moving around which makes it prone to drift, the proposed tracker is able to render the sequence completely without shifting to nearby objects. The sequence *quadrocopter* in Fig. 5.6(b) contains many challenges like fast motion, motion blur, camera motion and appearance changes. It is observed that FCT, DSST, EDFT, KCF, DAT trackers lose the target when quadrocopter moves sideways, but the proposed method achieves good tracking results due to combined approaches. Also, it can be observed that the gradient feature produces better results than channel coded representations and classifier based approach, thereby outweighing the gradient feature than other features. In Fig. 5.6(c), walking person is the target in *jacket* sequence which includes challenges like scale variation and low contrast. For this sequence, DSST and proposed tracker are proven to be more efficient compared to other algorithms. The *birds* sequence in Fig. 5.6(d) video include two people walking in the area surrounded by trees as background. The target moves suddenly when a flock of birds passes nearby and other trackers fail due to sudden pose changes. However, the proposed tracker can track the complete sequence irrespective of changes in appearance and background clutter.

5.2.3 Quantitative analysis

The comparison of proposed method with well known state-of-the-art algorithms has been illustrated on the basis of three criteria (Wu *et al.*, 2013): they include Average Center Location Error (ACLE), Distance Precision (DP) and Overlap Precision (OP). Deviation of a tracker from the ground truth location is depicted through the center location error graph as shown in Fig. 5.7. The ideal plot is expected to be closer to

the x-axis, indicating zero center location error. The proposed tracker produces CLE graph closer to the x-axis as compared to state-of-the-art trackers.



(a) *depthwise crossing* **(b)** *hiding* **(c)** *quadrocopter*

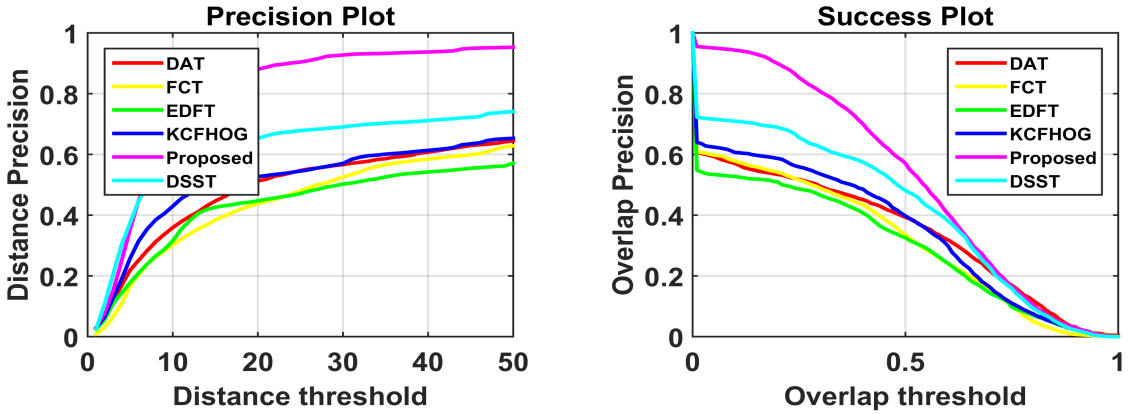
(d) *ragged* **(e)** *selma* **(f)** *trees1*

(g) *boat1* **(h)** *boat2* **(i)** *crouching*

Figure 5.7: Center Location Error plots of the proposed tracker against recent state-of-the-art trackers using challenging sequences like *depthwise crossing*, *hiding*, *quadrocopter*, *ragged*, *selma*, *trees1*, *boat1*, *boat2* and *crouching* image sequences.

Table 5.1: The tracking results of proposed tracker with five state-of-the-art methods using 17 infrared image sequences from LTIR dataset. The values are represented in triplet form: i.e. {distance precision score, PASCAL overlap score, average center location error}. The proposed method outperforms the compared algorithms in terms of overlap precision score, distance precision score, average center location error and are highlighted in **bold**.

Sequence	FCT	EDFT	DAT	KCF(HOG)	DSST	Proposed
birds	{44,73.7,21.2}	{21.8,31.4,56.4}	{38.1,54.0,37.8}	{24.4,43.7,35.2}	{55.9,55.5,28.6}	{**65.9,87.0,15.9**}
boat1	{99.2,**60**,6.5}	{4.8,4.8,329.2}	{97.2,53.1,8.3}	{18,16.4,136.3}	{99,59.2,**5.3**}	{**100**,44.8,6.3}
boat2	{96.3,42.1,7.0}	{90.4,58.6,16.9}	{97.7,51.7,6.2}	{**99,58.7**,5.3}	{98.9,45.7,5.7}	{97.5,38.8,**4.2**}
crouching	{3.8,21.0,48.2}	{62.7,59.2,25.7}	{36.7,39.8,30.6}	{33.9,29.7,49.3}	{**63.7,60.8**,24.9}	{60.8,60.3,**24.7**}
depthwise-crossing	{13.0,11.1,114.5}	{3.2,3.5,161.9}	{13.5,13.5,91.3}	{49,34,29.4}	{75.6,75.4,18.6}	{**100,100,4.4**}
dog	{2.1,0,123.3}	{75.9,10.8,13.0}	{54.3,8.6,21.6}	{**100**,19.5,5.5}	{**100,26.0,5.3**}	{**100**,15.2,**5.3**}
garden	{15.0,7.6,102.6}	{13.4,11.6,175.0}	{2.0,0.1,134.8}	{4.2,2.6,192.2}	{29.8,22.1,71.2}	{**100,89.2,5.8**}
hiding	{67.0,**66.7**,18.3}	{55.3,54.1,16.0}	{45.8,46.6,36.2}	{50.5,46,35.2}	{54.1,46.0,30.8}	{**78.2**,48.0,**14.2**}
jacket	{6.9,7.2,149.4}	{5.5,6.3,158.3}	{**51.0,53.4,30.1**}	{6.2,2.6,176}	{43.9,37.2,91.6}	{40.1,18.3,107.0}
quadrocopter	{6.1, 7.8,158.4}	{10.1,10.6,176.7}	{0.5,0.5,215}	{8.9,7.3,224.7}	{5.6,6.1,222.1}	{**89.3,78.0,16.1**}
ragged	{12.0,11.2,262.0}	{28.4,28.0,110.8}	{0.4,0.2,286}	{25.2,25.2,77.39}	{33.2,30.3,71.6}	{**94.1,36.8,10.5**}
running rhino	{**100**,35.6,9.9}	{**100**,13.8,9.9}	{**100**,77.7,5.5}	{**100**,85.5,4.5}	{**100,90.1,2.7**}	{98.2,48.3,8.4}
saturated	{96.3,94.4,6.5}	{**100,100**,10.8}	{98.6,96.7,**6.2**}	{**100,100**,8.9}	{**100,100,5.4**}	{94.0,90.8,7.8}
selma	{2.5,0,113.6}	{50.6,48.9,60.8}	{28.5,4.2,77.9}	{77.4,71.0,9.6}	{49.7,45.9,62.9}	{**100,23.8,8.7**}
soccer	{52,27.7,18.9}	{88.1,**68.3,8.9**}	{88.5,63.6,12.4}	{**85.8**,40.1,12.9}	{86.7,34.5,13.4}	{87.7,68.1,10.2}
street	{27.9,5.8,45.7}	{16.8,12.2,120.6}	{16.8,4.0,121.6}	{**100**,81.3,3.4}	{**100**,70.9,**3.9**}	{**100,88.9**,5.0}
trees1	{**100,96.9,4.3**}	{38.4,36.8,78.9}	{**100**,99.5,3.7}	{12.7,11.1,186.4}	{13.0,11.1,184.4}	{**100**,75.3,4.7}



(a) *Precision plot* **(b)** *Success plot*

Figure 5.8: Quantitative analysis of the proposed tracker and top 5 state-of-the-art trackers on 17 sequences of LTIR dataset. The plots are generated for one pass evaluation (OPE) running it once for given starting location. The proposed method achieves greater success rate as compared to the other investigated trackers.

To quantify the tracking methods graphically, precision and success plots are employed as shown in Fig. 5.8(a) and Fig. 5.8(b) respectively. It is evident from Fig. 5.8 and Table 5.1 that, the proposed tracker outperforms rest of the trackers in terms of

average center location error, precision score and overlap score. However, the proposed method runs at average of 6 frames per second.

5.3 Summary

In this chapter, a detection based tracking method has been presented that combines discriminative and generative approaches. The gradient feature and channel coded feature maps under kernelized correlation filter framework have been adaptively combined to get the object location. In parallel, AdaBoost classifier has been trained with object and background patches to classify pixels in every frame. The object has been localized in successive frames by performing mean shift procedure on the detected region. The optimum target location has been chosen that maximizes NCC score between the target and past history object model. Furthermore, the scale of target has been estimated in every frame using Lucas-Kanade optimization procedure. The combined approach takes advantage of individual techniques to make the proposed algorithm more resistant to drift. The proposed algorithm has been evaluated using 17 challenging videos from LTIR datasets and shown outstanding performance among state-of-the-art techniques. The drawback of proposed method is the timing complexity, which needs to be overcome by optimized hardware. In addition, there are certain disadvantages of thermal cameras which can be overcome by combining visual camera output with the thermal camera output for tracking applications. Tracking using combined imaging mode may resolve day time and night time tracking issues.

Chapter 6

VEHICLE COUNTING

Traffic control has become an essential part of the intelligent transport system due to ever growing society, number of roads and vehicles. Several types of research have been conducted for traffic management applications based on image and video processing approaches. They include detection/recognition of vehicles, analysis of vehicle speed, generating trajectory of vehicles, counting the vehicles, analysis of traffic congestion in real-time and automatic detection of accidents, and so on. Recently, video based traffic management system has become popular due to availability of low-cost cameras and low-cost embedded devices.

Real-time videos pose several challenges for automated traffic analysis system. The difficulties encountered by computerized system include the presence of shadows, illumination changes, occlusion of vehicles, environmental variations such as rain, fog, cloudy, dust, etc., which frequently deteriorate the performance. Despite many dedicated efforts, an accurate method for vehicle counting under complex environment is still far from being achieved. Due to the extensive utilization of cameras in urban transport systems, the surveillance video has become one of the central data sources. Also, real-time traffic management system has become popular recently due to the availability of handheld/mobile cameras. In this work, we present a video-based vehicle counting process using highway traffic videos captured using hand-held cameras. The processing of a video is achieved in three stages such as object detection using YOLO (You Only Look Once), tracking with correlation filter, and rule-based counting. YOLO produced a remarkable result in the object detection area, and correlation

filters achieved greater accuracy with competitive speed in tracking. In this chapter, we develop multiple object tracking with correlation filters using the bounding boxes generated by YOLO framework. Experimental analysis using real video sequences reveals that the proposed technique can detect, track and count vehicles precisely. In this chapter, we discuss the vehicle counting application for highway videos. Section 6.1 details about the object detection steps, tracking algorithm and the proposed technique to count the vehicles. Experimental analysis and discussion are provided in section 6.2 followed by summary in section 6.3.

6.1 Object detection and tracking

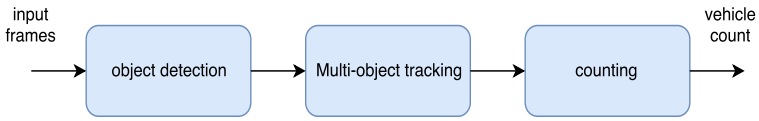


Figure 6.1: General block diagram of vehicle counting

Fig. 6.1 portrays the block diagram of vehicle counting process that include object detection, multi-object tracking and rule based counting. The real-time object detection is an active area in the computer vision field, and abundant researches have been proposed in the literature. At first, Haar features based cascaded Adaboost classifier has been proposed for face detection (Viola *et al.*, 2001). Later, Dalal *et al.* (2005) proposed Histogram of Gradient (HoG) based Support Vector Machine classifier to detect the pedestrians. Deformable Parts Model (DPM) has become attractive to identify the object using HoG and part based techniques (Felzenszwalb *et al.*, 2010). Recently, deep learning based approaches have been widely used due to availability of Graphical Processing Units (GPUs) and a huge amount of datasets. The techniques proposed by (Ren *et al.*, 2015) (Girshick *et al.*, 2015) respectively use CNN features with a sliding window or selective search method which is a time-consuming process. However, a robust method YOLO (Redmon *et al.*, 2016) treats the object detection as a regression problem to map pixels into bounding boxes with class probabilities. Moreover, it computes everything in a single evaluation, as a result, it runs in real-time.

6.1.1 Object detection

Motivated by the generalization property, performance accuracy, and speed of YOLO (Redmon *et al.*, 2016), we accommodate in the proposed work to serve the detection purpose. The main steps of YOLO are explained as follows:

- The input image divides into M x M grids, and that cell is responsible for an object if the center of an object falls into that cell.
- Each grid predicts B bounding boxes along with confidence scores. The score reflects how confident that the box contains an object.
- Each bounding box is represented using 5 predictions. i.e., [x, y, w, h, confidence score], where (x, y) denotes the center of box relative to the border of cell, (w, h) represents width and height relative to the image, confidence score represents intersection over union (IOU) between predicted box with the ground-truth boxes.
- YOLO has been trained using PASCAL VOC dataset and can predict 20 classes such as bicycle, boat, car, bus, person, motorbike, etc. Ultimately, the confidence score of bounding box and the class probabilities are multiplied to get final score that predicts the probability that bounding box has a particular object in it. Thus, each grid cell outputs class probabilities C.
- The network has 24 convolution layers followed by two fully connected layers.
- For 13×13 grid cells, each cell predicting 5 bounding boxes with 20 classes produce 845 boxes. Finally, boxes with more than 30% score is retained to estimate the objects in whole image. The flow of YOLO is given in Fig. 6.2.

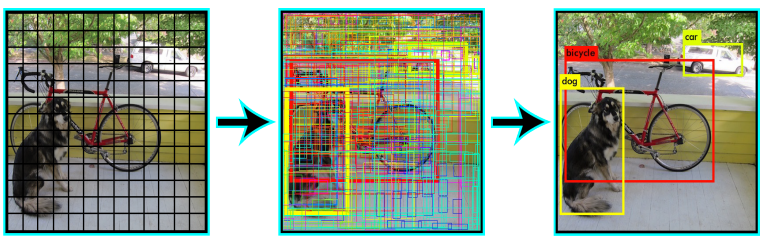


Figure 6.2: YOLO object detection process (Redmon *et al.*, 2016).

6.1.2 Vehicle tracking

In this context, tracking is the process of obtaining the location of moving vehicles in every frame of a video. The tracker generates trajectory of a vehicle starting from the given bounding box. For tracking experiments, the bounding box can be either user-specified or output of the object detector. The proposed work acquires initial bounding box using an object detector which requires vehicles on road as the desired object. Additionally, it exploits CF-based tracker (Danelljan *et al.*, 2014) to follow the vehicles. The successfulness of CF tracker to track multiple objects (Yang *et al.*, 2016) in real-time has motivated to use in the proposed method. Moreover, scale estimation is a crucial part of vehicle tracking system due to significant variation of vehicle's size which is accomplished by CF tracker. The block diagram of CF tracker is presented in Fig. 6.3. CF is trained using HoG features of vehicle data collected online using Gaussian template as the desired output. Thus, for each input sample and corresponding Gaussian output, the problem is formulated to create the filter template to obtain least error as:

$$\underset{\mathbf{h^l}}{\operatorname{argmin}} \left\| \sum_{l=1}^{k} \mathbf{h^l} * \mathbf{x^l} - \mathbf{y} \right\|^2 + \lambda \sum_{l=1}^{k} \left\| \mathbf{h^l} \right\|^2, \tag{6.1}$$

where $\mathbf{h}$ is the filter template in the spatial domain, λ is the regularization parameter, $\star$ denotes the convolution operation. The solution to Eq. (6.1) is obtained in the frequency domain as

$$\mathbf{H^l} = \frac{\mathbf{Y} \odot \mathbf{X^l}}{\sum_{l=1}^{k} \mathbf{X^l} \odot \mathbf{X^l} + \lambda}, \tag{6.2}$$

where $\mathbf{Y}$ represents Discrete Fourier Transform of $\mathbf{y}$, $\mathbf{X}$ denotes Discrete Fourier Transform of $\mathbf{x}$, $\odot$ refers element-wise multiplication. In order to adjust to recent appearances, the filter template is updated in every frame. Accordingly, the numerator $\mathbf{N^l_t}$ and denominator $\mathbf{D_t}$ of Eq. (6.2) are updated respectively as

$$\mathbf{N^l_t} = (1 - \eta)\mathbf{N^l_{t-1}} + \eta \mathbf{Y_t} \mathbf{X_t}^{\mathbf{l}}, \tag{6.3}$$

$$\mathbf{D_t} = (1-\eta)\mathbf{D_{t-1}} + \eta \sum_{l=1}^{k} \mathbf{X^l{}_t X^l{}_t}, \tag{6.4}$$

where η is learning rate fixed at 0.025. In subsequent frame, a rectangular patch $\mathbf{z}$ ($\mathbf{Z}$ in the frequency domain) is cropped from the current location and convolved with the filter template in the frequency domain to generate the correlation output as

$$\mathbf{o} = \Im^{-1} \left\{ \frac{\sum_{l=1}^{d} \mathbf{N^l{}_t} \odot \mathbf{Z}}{\mathbf{D_t} + \lambda} \right\}. \tag{6.5}$$

The peak value of $\mathbf{o}$ determines the location of target in the present frame. The size of target is estimated in every frame using 1-D correlation filter, which is trained using 33 different scaled versions of the object. The size of target in the current frame is found by searching a window with maximum correlation score among generated scaled patches. I-D scale filter is updated in every frame with learning parameter $\beta = 0.02$.

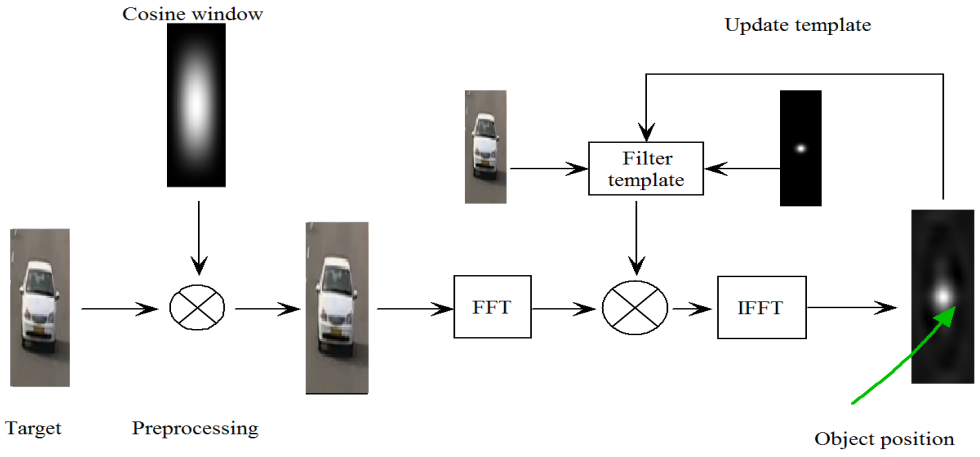


Figure 6.3: Block diagram of correlation filter based tracking

6.1.3 Vehicle counting

The proposed work combines the object detection unit with correlation filter based tracker to count the traffic data. The flowchart of proposed method is given in Fig. 6.4. In existing methods, background subtraction (BS) and blob tracking approaches have been utilized to detect the moving vehicles, and achieve good accuracy for fixed cameras and simple background only. However, BS methods produce strange results for unbalanced/shaking hand-held cameras. Hence, we utilize the state-of-the-art YOLO object detection framework to recognize the vehicles in video frames, followed by tracking using correlation filter. The sample frames of four different scenes are displayed in Fig. 6.5, where the videos have been recorded.

A part of the road area is cropped from the first frame and this region is considered as the entry window for all vehicles. The borders of image frame act as an exit line for every vehicle. Fig. 6.6(a) depicts a sample frame and manually selected entry window. The cropped area is termed as entry window and is displayed in Fig. 6.6(b). YOLO object detection algorithm is applied on entry window, and each detection is used to begin new track by assigning to a correlation filter after validating the object. All detections except **car**, **bus**, **motorbike**, **bicycle** are discarded from the detection process. The detection outputs are displayed in Fig. 6.6(c), and each object is denoted as ${OB_t}^j$. Thus, ${OB_t}^j$ denotes the j^{th} object detected in the t^{th} frame using YOLO framework. Let t, $t-1$, and $t+1$ denote the present, previous, and next frame respectively. Let ${CF_t}^i$ denotes i^{th} vehicle being tracked by the correlation filter in the t^{th} frame. The overlap between two bounding boxes is defined as the ratio of intersection over union (IOU) and is given by

$$O = \frac{OB \cap CF}{OB \cup CF}, \tag{6.6}$$

where OB denotes the object bounding box and CF symbolizes the tracked bounding box. The overlap factor O defines how well two bounding boxes overlap each other with 0 being no overlap and 1 indicates complete overlap. Initial step is to detect the new vehicles on the road and assigning to independent trackers. In addition, already detected vehicles are discarded by assigning to multiple trackers. Each tracker tracks the assigned vehicles till it reaches the other end of the frame. Finally, count is incremented (total vehicle count and individual count) based on pre-defined rules.

The flowchart depicted in Fig. 6.4 explains the various steps involved in the counting process. Accordingly, the following states are identified:

- **Track**: The object detected in a frame may correspond to one or more tracked bounding boxes. If the overlap between ${OB_t}^j$ and ${CF_t}^i$ is higher than the predefined threshold τ (in this work, $\tau = 0.3$), then the corresponding object bounding box is already assigned to a CF tracker and condition of the j^{th} vehicle is identified as **track** state. The size of vehicle progressively increases in every frame due to movement towards the camera; hence scale adaptation is very essential. Thus, the correlation filter locates each vehicle precisely due to its high efficiency and scale adaptation property. The tracker locates the vehicle in every frame and belong to active trackers. The correlation filters stop updating when the vehicle disappears from camera view. Subsequently, they are added to the list of passive trackers after removing from that of active trackers.

- **Detection**: If the overlap between ${OB_t}^j$ and ${CF_t}^i$ is less than the predefined threshold τ (in this paper, $\tau = 0.3$), then ${OB_t}^j$ is considered to be a distinct object which is detected in the frame t. In this condition, the tracked object is isolated from the detected object by an adequate distance. This state is identified as **detection** state. Further, the detected object is assigned to a new correlation filter based tracker to initiate the tracking process.

- **Termination**: If the coordinates of tracker corresponding to each vehicle reaches the border of frame, this state is named as **termination**. If the object is occluded, it reappears in the scene after a short period. However, the object disappears completely as a result of vehicle moving out of camera view. Consequently, the vehicle count is incremented. The corresponding tracker is removed from the list of active trackers and added to passive trackers. The vehicle count for each category is also incremented independently based on entity type.

- **Target Lost**: The tracking accuracy is proved to be high using correlation filter (Danelljan *et al.*, 2014). However, it may lose target when the smaller vehicle is occluded by larger vehicle. This state is referred as **target lost**. The failure of tracking is detected, when the bounding box does not move in any direction or moves in abnormal direction. To resolve this problem, the following assumptions are considered in the proposed work. They include (i)

all the vehicles are assumed to move in single direction (ii) if the tracking of a vehicle is lost due to occlusion or fast motion, then the corresponding vehicle will reach the boundary. Accordingly, the corresponding tracker is terminated, and the vehicle count is incremented.

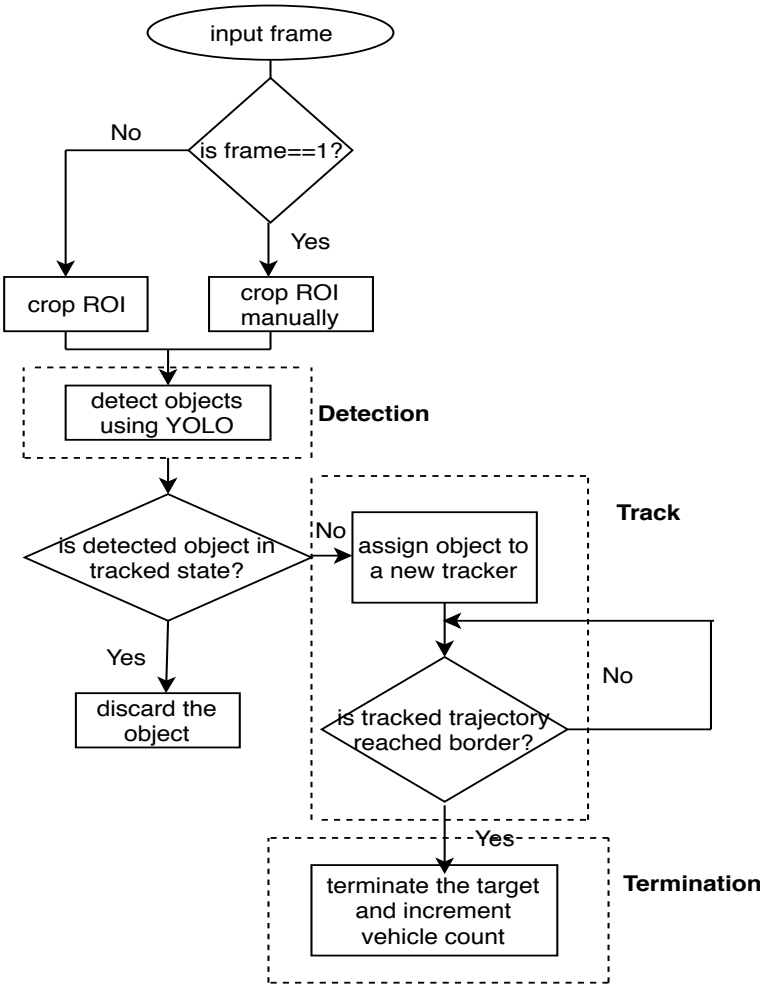


Figure 6.4: Flowchart of the proposed vehicle counting process.

6.2 Experimental analysis and discussion

6.2.1 Setup

The proposed algorithm is implemented using OpenCV 3.2 and PYTHON in a machine with Intel(R) Core i5-5200U, CPU of 2.20GHz processor and 8GB

RAM.

6.2.2 Datasets

For vehicle counting experiments, we prepared the video datasets using a mobile device with 13MP camera. These videos are acquired from the over-bridge in a highway using camera facing downwards with different illumination conditions and shadow effects. The videos are not stable since they have been captured by hand (not fixed). All videos have 1920 x 1080 resolution in RGB .mp4 format. Additionally, the videos contain complex background such as shaking plants, crossing pedestrians, surrounding buildings, trees, birds, and waving flags.

To initiate the counting process, a small road section is manually cropped in the first frame. For our dataset, a simple background subtraction algorithm often fails to extract the moving vehicles accurately. In addition, shadows and variations of illumination degrade the performance of background subtraction algorithm. Therefore, we employed a robust object detector, YOLO to detect and classify the moving vehicles in the entry window. Thus, the cropped region acts as an entry window where the tracking trajectory is initiated. Whereas, the border of frame acts as an exit line for all vehicles where tracking trajectory is terminated by incrementing the count. The processing time of YOLO for the image shown in Fig. 6.6(c) using the above mentioned CPU based machine is 1.5 sec. Also, that of the correlation filter based tracker is 0.013 sec for two objects. In spite of high detection rate, YOLO takes high processing time. Hence, to reduce the time, we execute YOLO after every M frames. (in this work M=5, depends on traffic density)

6.2.3 Quantitative and qualitative analysis

The experimental analysis includes comparison of vehicle count obtained using the proposed method with the manual count. For quantitative analysis, precision and recall evaluation metrics are considered (Wolf *et al.*, 2006). The precision (P) and recall (R) are defined as follows:

$$\mathrm{P} = \frac{\text{No of correctly detected bounding box}}{\text{No of groundtruth bounding box}} \tag{6.7}$$

and

$$\mathrm{R} = \frac{\text{No of correctly detected bounding box}}{\text{No of detected bounding box}}. \tag{6.8}$$

Thus, precision tells about false alarms while recall gives the information about how many of detected bounding boxes are correct. High value, close to 1 is expected for ideal systems. The vehicle is said to be detected if the overlap of detected bounding box and ground-truth bounding box is greater than 0.5. F-score measures harmonic mean of precision and recall as

$$\mathrm{F} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}}. \tag{6.9}$$

Figure 6.5: Sample locations of video used for vehicle counting. The videos are acquired using the hand-held mobile camera taken from the over-bridge. Four different locations are chosen to test the accuracy of the proposed method.

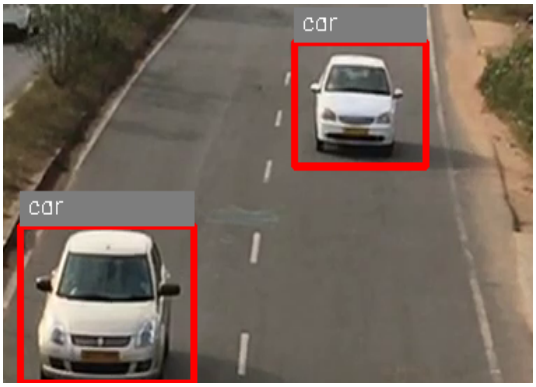
Similarly, counting accuracy is computed as

$$\text{Accuracy} = \frac{\text{No of correct detections}}{\text{No of groundtruth detections}}. \tag{6.10}$$

Table 6.1 provides the details of videos, precision, recall, F-measure, ground-truth count and vehicle count obtained using the proposed method. To test the efficiency of the proposed method, seven videos have been considered with low, medium and high traffic conditions. From Table 6.1, it is clear that the proposed method achieves 95.9% counting accuracy. Fig. 6.7(a) depicts 575^{th} frame of video 2.mp4. The vehicles currently in **track** state are shown in red. In this method, traffic density of small, medium and large vehicle categories are also provided. A sample frame is displayed in Fig. 6.7(b) with the count of different classes.

(a) Manually selected region of interest (ROI)

(b) Cropped ROI



(c) Object detection using YOLO framework on entry window (ROI)

(d) Object detection using YOLO framework on frame

Figure 6.6: Illustration of the proposed vehicle counting process.

Table 6.1: Vehicle count of obtained using the proposed method and manual count for hand recorded highway videos

Video Sequence	No of Frames	Total No. of Vehicles in the ground-truth	Total No of Vehicles detected using the proposed method	Missing or Multiple detection or error	Precision (%)	Recall (%)	F-score (%)	Counting accuracy (%)
1.mp4	899	10	10	0/0/0	100	100	100	100
2.mp4	715	16	15	1/0/1	93.7	100	96.7	93.7
3.mp4	845	17	18	0/1/1	100	94.4	97.1	94.4
4.mp4	3598	58	55	3/0/3	94.8	100	97.3	94.8
5.mp4	2100	25	23	2/0/2	92	100	95.8	92.0
6.mp4	5528	67	67	1/1/2	98.5	98.5	98.5	97.0
7.mp4	2999	28	28	0/0/0	100	100	100	100

(a) 575^{th} frame of video 2.mp4. Total no of vehicles passed are displayed at the top corner. Vehicles under tracked state are shown using red bounding box.

(b) 860^{th} frame of video 1.mp4. Total no of vehicles passed are displayed at the top corner. Traffic count is also classified as small (LOWCOUNT), medium (MEDIUMCOUNT) and large (HIGHCOUNT) categories. Vehicles in the tracked state are shown using red bounding box.

Figure 6.7: Sample frames of vehicle counting algorithm.

6.3 Summary

In this chapter, counting of vehicles in a mixed traffic condition has been proposed. We exploited YOLO framework to detect the vehicles and correlation

filters to track precisely. The traffic density of selected videos varies from low to high, and the proposed method counted the vehicles accurately. The advantage of proposed method is that it can be generalized to any kind of road videos captured using the hand-held mobile camera. Moreover, YOLO can also distinguish the vehicle classes, hence counting is also accomplished for different categories to analyze the count of each vehicle type in a traffic video.

Chapter 7

CONCLUSIONS AND FUTURE WORK

This chapter summarizes the contributions of thesis, and suggests potential directions for future research. Section 7.1 presents the contributions of thesis in tracking area, and conclusions are given in 7.2 followed by possible future research directions in section 7.3.

7.1 Contributions

In this thesis, we have suggested an improved tracking technique for illumination invariant drift free tracking of given object in RGB videos, tracking of an object in thermal infrared videos and an application of tracking for counting vehicles. The proposed approaches have been illustrated using challenging videos and presented better accuracy compared to baseline trackers. The contributions of thesis are listed below.

In chapter 3, we reviewed the well known median flow tracker for tracking an object using optical flow technique. We found that MFT is based on frame-to-frame tracking method, depends on pixel values in the successive frames for efficient tracking. However, it is realized that MFT slips during abrupt illumination variation which prompts the tracker to drift. Hence, we suggested prefixing a photometric normalization method before tracking begins. Several

effective normalization techniques have been discussed to handle illumination caused problems by either extracting the reflectance component or maintaining the uniform light conditions.

In chapter 4, we studied the recent correlation filter based tracker and suggested enhancements to improve the accuracy. The correlation filters learn from spatial features which makes tracker to drift due to occlusion, fast motion and object deformation. Hence, the proposed study incorporates two complementary trackers (i.e., discriminative and generative) to obtain the location of the object in every frame. In addition, the dynamic learning rate has been used to overcome occlusion. Finally, we tested the proposed method using challenging video sequences and have shown better accuracy compared with the baseline trackers.

In chapter 5, we extended the correlation filter based tracker to track an object in infrared imagery. The proposed approach combines spatial features with intensity features in a correlation filter and AdaBoost classifier framework respectively to retrieve probable locations. Ultimately, the actual position has been determined by applying a generative technique. A novel scale estimation is proposed to track scale in every frame. The combined approach has been tested on several challenging infrared videos to show considerable improved accuracy among the state-of-the-art trackers.

In chapter 6, we presented a novel technique to count the vehicles in a highway traffic video. This method exploits recent YOLO framework to detect the vehicle classes and correlation filter for multi-object tracking purpose. Finally, a simple rule-based technique is employed to estimate the vehicle count. Also, we generated video datasets for counting vehicles on highway videos. The count is then compared with the manual count to show its accuracy.

7.2 Conclusions

In this thesis, video tracking algorithms have been analyzed by providing an outline of state-of-the-art as well as the proposed techniques. Each chapter described the possible extensions to state-of-the-art trackers. Thus, median flow tracker (generative approach) and kernelized correlation filter based tracker (discriminative method) have been selected as baseline trackers. An illumination

invariant median flow tracker has been suggested by incorporating photometric normalization techniques. The experimental analysis showed that the modified tracker outperforms the base tracker, also some of the recent state-of-the-art trackers in terms of robustness and accuracy.

Moreover, an approach to deal with occlusion, fast motion, and illumination variations has been presented. This is accomplished using kernelized correlation filter as a base tracker with novel feature selection criteria and adaptive learning rate scheme. Further, switching to a complementary tracker based on predefined rules has increased the accuracy. Similarly, a robust tracking achieved in thermal infrared videos using complementary trackers in parallel. The combined approach has been optimized to obtain the best solution. The experiments revealed that the proposed method is accurate in locating the target.

Finally, an application of video tracking has been discussed to count the vehicles in highway traffic. The combination of object detector and correlation filter tracker has numbered vehicles close to human accuracy. Comprehensive experimental evaluation has been conducted to test the algorithms mentioned.

7.3 Future work

A significant progress has been achieved in the field of tracking in the last decade. However, there is a need of single system that works for real time situation. The thesis has discussed illumination invariant techniques for median flow tracker. However, several other recent trackers fail during sudden illumination changes. Thus, the proposed methods can be easily extended to the illumination sensitive trackers. Also, the reflectance estimation or illumination constancy algorithms have utilized grayscale images. Instead, the color images can be used to provide better accuracy.

A number of generative and discriminative techniques are available in the literature. Thus, the various combinations can be tested to provide better accuracy with fewer computations in both RGB and infrared videos.

Highway traffic video in developing countries is always challenging due to following reasons: (i) the videos contain unexpected movements or crossings of humans in the roads (ii) variety of vehicle types are observed which leads to

multiple detections of vehicles (iii) although, the lane is meant for single directions, vehicles are observed moving in the other direction also. Hence, all these problems need to be discussed in the future work. In addition, the single lane has been considered to count the vehicles in literature, which can be extended to multi paths without increasing the computational complexity. Also, an accurate count of individual vehicle type helps to understand the vehicle density in roads. However, recently vehicle number has been increased to the large extent resulting in traffic congestion, accidents, etc. Hence, an intelligent system is needed to understand the traffic condition in the future.

Bibliography

Adam, A., **Rivlin, E.**, and **Shimshoni, I.**(2006). "Robust fragments-based tracking using the integral histogram". *IEEE Computer Society Conference on Computer vision and pattern recognition, 2006* , **1**, 798–805.

Armato, A., **Lanat, A.**, and **Scilingo, E. P.**(2013). "Comparitive study on photometric normalization algorithms for an innovative, robust and real-time eye gaze tracker". *Journal of real-time image processing*, **8**(1), 21–33.

Asha C., **Narasimhadhan A.**(2017). "Robust infrared target tracking using discriminative and generative approaches". *Infrared Physics & Technology*, **85**, 114–127.

Asvadi, A., **Mahdavinataj, H.**, **Karami, M.**, and **Baleghi, Y.**(2013). "Incremental Discriminative Color Object Tracking". *In International Symposium on Artificial Intelligence and Signal Processing*, 71–81.

Avidan, S.(2007). "Ensemble tracking". *IEEE transactions on pattern analysis and machine intelligence*,**29**(2).

Avidan, S.(2004), "Support vector tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**, 1064–1072.

Avidan, S.(2006). "Spatialboost: Adding spatial reasoning to adaboost". *European Conference on Computer Vision Springer*, 386–396.

Babenko, B., **Yang, M.H.**, **Belongie, S.**(2011). "Robust object tracking with online multiple instance learning". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **33**, 1619–1632.

Baker, S. and **I. Matthews**(2004). "Lucas-kanade 20 years on: A unifying framework". *International journal of computer vision*, **56**(3), 221–255.

Barcellos, P.,**Bouvi, C.**, **Escouto, F. L.**, and **Scharcanski, J.**, (2015). "A novel video-based system for detecting and counting vehicles at user-defined virtual loops". *Expert Systems with Applications*, **42**(4), 1845–1856.

Berg, A., **J. Ahlberg**, and **M. Felsberg** (2015). "A thermal object tracking benchmark". *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).*

Berg, A., **Ahlberg, J.**, and **Felsberg, M.**(2016). "Channel coded distribution field tracking for thermal infrared imagery". *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 9–17.

Berg, A., **Ahlberg, J.**, and **Felsberg, M.** (2016). "Channel coded distribution field tracking for thermal infrared imagery". *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 9–17.

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

Bhaskar, P. K., and **Yong, S. P.**(2014). "Image processing based vehicle detection and tracking method". *In Computer and Information Sciences (ICCOINS), 2014 International Conference on*, 1–5.

Bolme, D. S., **J. R. Beveridge**, **B. Draper**, and **Y. M. Lui**(2010), "Visual object tracking using adaptive correlation filters", *Computer Vision and Pattern Recognition (CVPR),2544–2550.*

Briechle, K. *and* **U. D. Hanebeck***(2001), "Template matching using fast normalized cross correlation". Aerospace/Defense Sensing, Simulation, and Controls,****4387****, 95–102*

Chen, T. H., **Lin, Y. F.**, *and* **Chen, T. Y.***(2007). "Intelligent vehicle counting method based on blob analysis in traffic surveillance".* In Innovative Computing, Information and Control Second International Conference on*, 238–238.*

Chen, W.,**Er, M. J.**, *and* **Wu, S.***(2006). "Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain". IEEE Transactions on Systems, Man, and Cybernetics,* ***36****(2), 458–466.*

Chen, J., **Shan, S.**, **He, C.**, **Zhao, G.**, **Pietikainen, M.**, **Chen, X.**, *and* **Gao, W.***(2010). "WLD: A robust local image descriptor". IEEE transactions on pattern analysis and machine intelligence,* ***32****(9), 1705-1720.*

Collins, R. T., **Y. Liu**, *and* **M. Leordeanu***(2005). "Online selection of discriminative tracking features". IEEE transactions on pattern analysis and machine intelligence,***27***(10), 1631–1643.*

Comaniciu, D., **V. Ramesh**, and **P. Meer** (2000). "Real-time tracking of non-rigid objects using mean shift". *IEEE Conference on Computer Vision and Pattern Recognition*, **2**, 142–149).

Cucchiara, R., **Piccardi, M.**, and **Mello, P.** (2000). "Image analysis and rule-based reasoning for a traffic monitoring system".. *IEEE Transactions on Intelligent Transportation Systems*, **1**(2), 119–130.

Dalal, N., and **Triggs, B**(2005). "Histograms of oriented gradients for human detection". *In Computer Vision and Pattern Recognition, IEEE Computer Society Conference on", **1**, 886–893.*

Danelljan, M., **F. S. Khan**, **M. Felsberg**, *and* **J. van de Weijer***(2014). "Adaptive color attributes for real-time visual tracking". Computer Vision and Pattern Recognition (CVPR),1090–1097.*

Danelljan, M., **G. Hager**, **F. Khan**, *and* **M. Felsberg***(2014). "Accurate scale estimation for robust visual tracking". British Machine Vision Conference, Nottingham*

Danelljan, M., **Hager, G.**, **Shahbaz Khan, F.**, *and* **Felsberg, M.** *(2015). "Learning spatially regularized correlation filters for visual tracking". In Proceedings of the IEEE International Conference on Computer Vision, 4310–4318.*

Dollar, P., **Z. Tu**, **P. Perona**, *and* **S. Belongie***(2009). "Integral channel features". 91-1.*

Dollár, P. *(2009). Piotr's Computer Vision Matlab Toolbox (PMT),* `https://github.com/pdollar/toolbox`.

Dong, X., **X. Huang**, **Y. Zheng**, **S. Bai**, *and* **W. Xu***(2014). "A novel infrared small moving target detection method based on tracking interest points under complicated background". Infrared Physics & Technology,* **65**, *36–42.*

Dong, X., **X. Huang**, **Y. Zheng**, **L. Shen**, *and* **S. Bai***(2014). "Infrared dim and small target detecting and tracking method inspired by human visual system". Infrared Physics & Technology,* **62**, *100–109.*

Du, S., *and* **Ward, R.***(2005). "Wavelet-based illumination normalization for face recognition". IEEE International Conference on Image Processing, **2**, II-954.*

Felsberg, M.*(2013), "Enhanced distribution field tracking using channel representations". Proceedings of the IEEE International Conference on Computer Vision Workshops, 121–128.*

Felsberg, M., **A. Berg**, **G. Hager**, **J. Ahlberg**, **M. Kristan**, **J. Matas**, **A. Leonardis**, **L. Cehovin**, **G. Fernandez**, **T. Vojir**, et al.*(2015). "The thermal infrared visual object tracking vot-tir2015 challenge results". Proceedings of the IEEE International Conference on Computer Vision Workshops.*

Felzenszwalb, P. F., **Girshick, R. B.**, *and* **McAllester, D.***(2010). "Cascade object detection with deformable part models".* In Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, 2241–2248.*

Felzenszwalb, P.F., **Girshick, R.B.**, **McAllester, D.**, **Ramanan, D.***(2010), "Object detection with discriminatively trained part-based models", IEEE transactions on pattern analysis and machine intelligence,* ***32****, 1627–1645.*

Gade, R. *and* **T. B. Moeslund***(2014). "Thermal cameras and applications: a survey". Machine vision and applications,* **25***(1), 245–262.*

Girshick, R.*(2015), "Fast r-cnn", In Proceedings of the IEEE international conference on computer vision, 1440–1448.*

Grabner, H., **M. Grabner**, *and* **H. Bischof***(2006). "Real-time tracking via on-line boosting". BMVC,* ***1****(5), 6.*

Grabner, H., *and* **Bischof, H.** *(2006). "On-line boosting and vision". IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* ***1****, 260–267.*

Grabner, H., **Bischof, H.***(2006), "On-line boosting and vision", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 260–267.*

Hadi, R. A., **George, L. E.**, **Mohammed, M. J.***(2017). A computationally economic novel approach for real-time moving multi-vehicle detection and tracking toward efficient traffic surveillance.* Arabian Journal for Science and Engineering*, 2017,* ***42****, 817–831.*

Han, H., **Shan, S.**, **Chen, X.**, *and* **Gao, W.***(2013). "A comparative study on illumination preprocessing in face recognition". Pattern Recognition,* ***46****(6), 1691–1699.*

Henriques, J. F., **R. Caseiro**, **P. Martins**, *and* **J. Batista***(2012), "Exploiting the circulant structure of tracking-by-detection with kernels". European conference on computer vision Springer, 702–715.*

Henriques, J. F., **R. Caseiro**, **P. Martins**, *and* **J. Batista***(2015). "High-speed tracking with kernelized correlation filters". IEEE Transactions on Pattern Analysis and Machine Intelligence,* ***37****(3), 583–596.*

Hare, S.,**Golodetz, S.**, **Saffari, A.**, **Vineet, V.**, **Cheng, M. M.**, **Hicks, S. L.**, and **Torr, P. H.**(2016). "Struck: Structured output tracking with kernels". *IEEE transactions on pattern analysis and machine intelligence*, **38**(10), 2096–2109.

He, Y.-J., **M. Li**, **J. Zhang**, and **J.-P. Yao**(2015). Infrared target tracking via weighted correlation filter. *Infrared Physics & Technology*, **73**, 103–114.

Huang, S. C., **Cheng, F. C.**, and **Chiu, Y. S**(2013), "Efficient contrast enhancement using adaptive gamma correction with weighting distribution", *IEEE Transactions on Image Processing*, **22**(3), 1032-1041.

Jang, H., **Won, I. S.**, and **Jeong, D. S.**(2014). "Automatic Vehicle Detection and Counting Algorithm". *International Journal of Computer Science and Network Security (IJCSNS)*, **14**(9), 99.

Jobson, D. J., **Rahman, Z. U.**, and **Woodell, G. A.**(1997). "Properties and performance of a center/surround retinex". *IEEE transactions on image processing*, **6**(3), 451–462.

Jonsson, E.(2008). "Channel-coded feature maps for computer vision and machine learning", Ph.D. thesis, Institutionen för systemteknik.

Kalal, Z., **Mikolajczyk, K.**, and **Matas, J.**(2010). "Forward-backward error: Automatic detection of tracking failures". *International Conference on Pattern Recognition (ICPR)*, 2756–2759.

Kalal, Z., **K. Mikolajczyk**, and **J. Matas**(2012). Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, **34**(7), 1409–1422.

Kamkar, S., and **Safabakhsh, R.** (2016). "Vehicle detection, counting and classification in various conditions". *IET Intelligent Transport Systems*, **10**(6), 406–413.

Khan, F.S., **Anwer, R.M.**, **Van De Weijer, J.**, **Bagdanov, A.D.**, **Vanrell, M.**, **Lopez, A.M.(2012)**. "Color attributes for object detection". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3306–3313.

Lamberti, F., **A. Sanna**, and **G. Paravati**(2011). "Improving robustness of infrared target tracking algorithms based on template matching". *IEEE Transactions on Aerospace and Electronic Systems*, **47**(2), 1467–1480.

Land, E. H., and **McCann, J. J.**(1971). "Lightness and retinex theory". *JOSA*, **61**(1), 1–11.

Lee, S. J., **G. Shah**, **A. A. Bhattacharya**, and **Y. Motai**(2012). "Human tracking with an infrared camera using a curve matching framework". *EURASIP Journal on Advances in Signal Processing*, **2012**(1), 1.

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

Liu, X., **Wang, Z.**, **Feng, J.**, and **Xi, H.**(2016). "Highway vehicle counting in compressed domain". *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3016–3024.

Li, Y. and **J. Zhu**(2014). A scale adaptive kernel correlation filter tracker with feature integration. *European Conference on Computer Vision Springer*, 254–265.

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

Maggio, E., **Cavallaro, A**, Video tracking: theory and practice, *In John Wiley & Sons*, 2011.

Mahalanobis, A., **Kumar, B. V.**, and **Casasent, D.**(1987). "Minimum average correlation energy filters". *Applied Optics*, **26**(17), 3633–3640.

Mei, X. and **H. Ling**(2009). "Robust visual tracking using l1 minimization". *IEEE International Conference on Computer Vision*, 1436–1443.

Moranduzzo, T., and **Melgani, F.**(2014). "Automatic car counting method for unmanned aerial vehicle images". *IEEE Transactions on Geoscience and Remote Sensing", **52**(3), 1635–1647.*

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

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

Nhat, V. Q., *and* **Lee, G.***(2014). "Illumination invariant object tracking with adaptive sparse representation". International Journal of Control, Automation and Systems, **12**(1), 195–201.*

Ning, J., **Zhang, L.**, **Zhang, D.**, *and* **Wu, C.***(2012). "Robust mean-shift tracking with corrected background-weighted histogram". IET computer vision,* ***6****(1), 62–69.*

Ochoa-Villegas, M. A.,**Nolazco-Flores, J. A.**, **Barron-Cano, O.**, *and* **Kakadiaris, I. A.***(2015). "Addressing the illumination challenge in two-dimensional face recognition: a survey". IET Computer Vision,* ***9****(6), 978–992.*

Phadke, G., **Velmurgan, R.***(2013). "Illumination invariant Mean-shift tracking". IEEE Workshop On Applications of Computer Vision (WACV), 407–412.*

Phadke, G., *and* **Velmurugan, R.***(2017). "Mean LBP and modified fuzzy C-means weighted hybrid feature for illumination invariant mean-shift tracking". Signal, Image and Video Processing,* ***11****(4), 665–672.*

Pornpanomchai, C.,**Liamsanguan, T.**, *and* **Vannakosit, V.***(2008). "Vehicle detection and counting from a video frame". In Wavelet Analysis and Pattern Recognition, ICWAPR'08. International Conference on,* ***1****, 356–361.*

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

Quesada, J., *and* **Rodriguez, P.***(2016). "Automatic vehicle counting method based on principal component pursuit background modeling".* In Image Processing (ICIP), 2016 IEEE International Conference on*, 3822–3826.*

Redmon, J., **Divvala, S.**, **Girshick, R.**, *and* **Farhadi, A.***(2016). "You only look once: Unified, real-time object detection".* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788.*

Ren, S., **He, K.**, **Girshick, R.**, *and* **Sun, J***(2015). "Faster R-CNN: Towards real-time object detection with region proposal networks".* In Advances in neural information processing systems*, 91–99.*

Rodriguez, M. D., **Ahmed, J.**, *and* **Shah, M.***(2008). "Action mach a spatio-temporal maximum average correlation height filter for action recognition". IEEE Conference on Computer Vision and Pattern Recognition, 1–8.*

Ross, D. A., **J. Lim**, **R.-S. Lin**, *and* **M.-H. Yang***(2008). "Incremental learning for robust visual tracking". International Journal of Computer Vision,* ***77****(1), 125–141.*

Salvi, G.*(2014). "An automated nighttime vehicle counting and detection system for traffic surveillance".* In Computational Science and Computational Intelligence (CSCI), 2014 International Conference on*,* ***1****, 131–136.*

Samuel, O.W.*,* **Asogbon, G.M.***,* **Sangaiah, A.K.***,* **Li, G.***(2017). Multi-technique object tracking approach-a reinforcement paradigm. Computers & Electrical Engineering.*

Santner J.*,* **Leistner C.***,* **Saffari A.***,* **Pock T.***,* **Bischof H.***(2010). "Prost: Parallel robust online simple tracking". IEEE Conference on Computer Vision and Pattern Recognition, 723–730.*

Sevilla-Lara, L. *and* **E. Learned-Miller***(2012). "Distribution fields for tracking". IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1910–1917.*

Struc, V.*, and* **Paveic, N.***(2011). "Photometric normalization techniques for illumination invariance". Advances in face image analysis: Techniques and technologies, 279–300.*

Struc, V.*(2012). The INface toolbox v2. 1 The Matlab Toolbox for Illumination Invariant Face RecognitionToolbox description and user manual. University of Ljubljana, 2012.*

Tan, X., *and* **Triggs, B.***(2010). "Enhanced local texture feature sets for face recognition under difficult lighting conditions". IEEE transactions on image processing,* ***19****(6), 1635–1650.*

Van De Weijer, J.*,* **Schmid, C.***,* **Verbeek, J.***,* **Larlus, D.***(2009). "Learning color names for real-world applications". IEEE Transactions on Image Processing,* ***18****, 1512–1523.*

Van Pham, H.*, and* **Lee, B. R.***(2015). "Front-view car detection and counting with occlusion in dense traffic flow", International Journal of Control, Automation and Systems,* ***13****(5), 1150–1160.*

Viola, P.*, and* **Jones, M.***(2001). "Rapid object detection using a boosted cascade of simple features".* In Computer Vision and Pattern Recognition Proceedings of the 2001 IEEE Computer Society Conference on*,* ***1****, I-I.*

Vu, N. S.*, and* **Caplier, A.***(2009). "Illumination-robust face recognition using retina modeling". 16th IEEE International Conference on Image Processing (ICIP), 3289–3292.*

Wang, H.*,* **Li, S. Z.***,* **Wang, Y.***, and* **Zhang, J.***(2004). "Self quotient image for face recognition". International Conference on Image Processing,* ***2****, 1397–1400.*

Wang, B., **Li, W.**, **Yang, W.**, *and* **Liao, Q.***(2011), "Illumination normalization based on weber's law with application to face recognition", IEEE Signal Processing Letters,* ***18****(8), 462–465.*

Wang, J. T., **D. B. Chen**, **H. Y. Chen**, *and* **J. Y. Yang***(2012). "pedestrian detection and tracking in infrared videos". Pattern Recognition Letters,* **33***(6), 775–785.*

Wang, X. *and* **Z. Tang***(2010). "Modified particle filter-based infrared pedestrian tracking". Infrared Physics & Technology,* **53***(4), 280–287.*

Wolf, C., *and* **Jolion, J. M.**,*(2006). "Object count/area graphs for the evaluation of object detection and segmentation algorithms". International Journal of Document Analysis and Recognition (IJDAR),* ***8****(4), 280–296.*

Wu, Y., **J. Lim**, *and* **M.-H. Yang***(2013). "Online object tracking: A benchmark".* Proceedings of the IEEE conference on computer vision and pattern recognition*, 2411–2418.*

Xia, Y., **Shi, X.**, **Song, G.**, **Geng, Q.**, *and* **Liu, Y.***(2016). "Towards improving quality of video-based vehicle counting method for traffic flow estimation". Signal Processing,* **120***, 672–681.*

Yang, F., **Lu, H.**, **Zhang, W.**, *and* **Yang, G.***(2012). "Visual tracking via bag of features". IET image processing,* ***6****(2), 115–128.*

Yang, Y., *and* **Bilodeau, G. A.***(2016). "Multiple Object Tracking with Kernelized Correlation Filters in Urban Mixed Traffic."*arXiv preprint arXiv:1611.02364*.*

Yu, Q., **Dinh, T. B.**, *and* **Medioni, G***(2008). "Online tracking and reacquisition using co-trained generative and discriminative trackers". In European conference on computer vision, 678–691.*

Zhang, K., **Zhang, L.**, *and* **Yang, M. H.***(2012). "Real-time compressive tracking". In European conference on computer vision, 864–877.*

Zhang, K., **L. Zhang**, *and* **M.-H. Yang***(2014). "Fast compressive tracking".* IEEE Transactions on Pattern Analysis and Machine Intelligence*,* **36***(10), 2002–2015.*

Zhang, T., **Fang, B.**, **Yuan, Y.**, **Tang, Y. Y.**, **Shang, Z.**, **Li, D.**, *and* **Lang, F.***(2009). "Multiscale facial structure representation for face recognition under varying illumination". Pattern Recognition,* ***42****(2), 251–258.*

PUBLICATIONS

International Journal Papers

1. **Asha, C. S.**, Narasimhadhan, A. V., Robust infrared target tracking using discriminative and generative approaches, *Infrared Physics & Technology*, 85, 114-127, 2017, **Elsevier Publisher, (SCI Indexed)**.
2. **Asha C. S.**, Narasimhadhan A. V.,Visual Tracking using Kernelized Correlation Filter with Conditional Switching to Median Flow Tracker, IETE Journal of Research, **Taylor and Francis Publisher**, **(SCI Indexed)**.
3. **Asha C. S.**, Narasimhadhan A. V., Enhanced Median Flow Tracker Based on Photometric Correction for Videos with Abrupt Changing Illumination, The International Arab Journal of Information Technology **(SCI Indexed)** (**Accepted**)
4. **Asha C. S.**, Narasimhadhan A. V., A Comparative Study of Illumination Invariant Techniques in Video Tracking Perspective, IETE Technical Review (**Communicated**)

International Conference Papers

1. **Asha C. S.**, Narasimhadhan A. V., Adaptive Learning Rate for Visual Tracking Using Correlation Filters. *Procedia Computer Science*, 89, 614-622, 2016.
2. **Asha C. S.**, Narasimhadhan A. V., Vehicle Counting for Traffic Management System using YOLO and Correlation Filter, IEEE Conecct, (**Accepted**)

National Conference Papers

1. **Asha C. S.**, Narasimhadhan A. V., Experimental evaluation of feature channels for object tracking in RGB and thermal imagery using correlation filter, *In Communications (NCC), 2017 Twenty-third National Conference on, IEEE*, March 2017, 1-6.

CURRICULUM VITAE

ASHA C S
Sanjay Nagar first cross
Opposite court
Belthangady Taluk
South canara, 574214
Email: asha.cs@rediffmail.com
asha.cs12@gmail.com
Mobile: +91-9901099630

Academic Records

1. M.Tech. in Digital Electronics and Communication, NMAMIT, Nitte. CGPA 8.33/10).
2. B.Tech. in Electronics and Communication Engineering, NMAMIT, Nitte. Aggregate 75%).

Research Interests

Computer Vision, Video Processing, Medical Image Processing.

Programming Languages

Matlab, OpenCV, Python.