SURFACE VESSEL TRACKING IN AIRBORNE INFRARED IMAGERY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AHMET ÇAKIROĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2019

Approval of the thesis:

SURFACE VESSEL TRACKING IN AIRBORNE INFRARED IMAGERY

submitted by **AHMET ÇAKIROĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Tolga Çiloğlu
Head of Department, **Electrical and Electronics Engineering** ________________

Prof. Dr. İlkay Ulusoy
Supervisor, **Electrical and Electronics Engineering, METU** ________________

Examining Committee Members:

Prof. Dr. Uğur Halıcı
Electrical and Electronics Engineering, METU ________________

Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering, METU ________________

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering, METU ________________

Prof. Dr. Alptekin Temizel
Informatics Institute, METU ________________

Assist. Prof. Dr. S. Esen Yüksel
Electrical and Electronics Engineering, Hacettepe University ________________

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Ahmet Çakıroğlu

Signature :

ABSTRACT

SURFACE VESSEL TRACKING IN AIRBORNE INFRARED IMAGERY

Çakıroğlu, Ahmet
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. İlkay Ulusoy

February 2019, 72 pages

Target tracking can be defined as continuously locating the object of interest in consequent images. Tracking surface vessels in infrared imagery is an exceptionally challenging case of visual target tracking. In a typical scenario both the target and imaging platform exhibit manoeuvring movement, causing the appearance of the target to change rapidly and significantly during the course of tracking. Furthermore there are cases where target actively attempts to avoid being tracked by firing hot flares to confuse the tracker or block the view of the tracker. In some cases target also cools itself down to background temperatures with special equipment to blend itself with the background and avoid being seen. In this thesis one of the popular general object tracking algorithms is improved by transfer learning and developing an occlusion detection mechanism. Discrimination power of the tracker is increased by transfer learning and occlusion detection capabilities enabled the tracker to reacquire the target after occlusion. Performance of proposed algorithm and other several distinguished target tracking algorithms are compared on our infrared surface vessel image dataset. Image dataset consists of synthetic images acquired during challenging naval combat scenarios and categorized by their respective challenges such as confusion, low intensity and occlusion. It was seen that the proposed algorithm had superior performance to other tested algorithms.

Keywords: Target tracking, Object Tracking, Visual Tracking, Infrared Tracking

ÖZ

HAVADAN ALINAN KIZILÖTESİ GÖRÜNTÜLERDE GEMİ TAKİBİ

Çakıroğlu, Ahmet
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. İlkay Ulusoy

Şubat 2019 , 72 sayfa

Hedef takibi, istenilen bir nesnenin ardışık görüntüler üzerinde sürekli olarak yerinin belirlenmesi olarak tanımlanabilir. Su üstü gemilerin takibi, hedef takibinin büyük derecede zorlu bir uygulamasıdır. Tipik bir senaryoda hedef ve görüntü alan platformun yaptığı manevralı hareketler, görüntünün hızlıca değişmesine sebep olmaktadır. Buna ek olarak hedefin takip edilmekten kaçınmak amacıyla havaya fırlattığı sıcak fişekler takip algoritmasını yantılmakta veya görüntüsünü kapatmaktadır. Bazı durumlarda ise özel ekipmanlar kullanılarak, hedef arka plan scaklığına kadar soğutulmakta ve görülmesinin engellenmesi amaçlanmaktadır. Bu tezde popüler hedef takip algoritmalarından bir tanesi öğrenim aktarımı ve görüntüde kaybolmanın tespiti yeteneğinin eklenmesi ile geliştirilmiştir. Öğrenim aktarımı işlemi ile takipçinin ayırt edicilik yetenği arttırılmış , görüntüde kaybolma tespiti ile ise hedef tekrar belirdiğinde takibe devam edilebilmesi sağlanmıştır. Önerilen algoritmanın ve literatürde kendini ispat etmiş diğer bir kaç algoritmanın başarımları kendi hazırladığımız bir kızılötesi gemi görüntüsü veri kümesi üzerinde test edilmiştir. Görüntü veri kümesi zorlayıcı deniz savaş ortamında elde edilmiş ve içerdikleri karıştırma, düşük parlaklık ve görüntüde kaybolma gibi zorluklara göre kategorilendirilmiş sentetik kızılötesi gemi görüntülerinden oluşmaktadır. Elde edilen sonuçlarda önerilen algoritmanın diğer algoritmalara göre üstün başarım gösteriği görülmüştür.

Anahtar Kelimeler: Hedef takibi, Nesne takibi, Görsel takip, Kızılötesi takip

To my family...

ACKNOWLEDGMENTS

I want to express my gratitude to my supervisor Prof. Dr. İlkay Ulusoy for her guidance, advice, criticism, encouragements and insight throughout my research.

I would like to give my special thanks to my friends for their encouragement, support and above all for their friendship throughout the study.

I would also like to thank my colleagues at TÜBİTAK SAGE for their moral, technical and academical support.

Finally, my deep and sincere gratitude to my family for their continuous and unparalleled love, help and support. I am grateful to my brother for always being there for me. I am forever indebted to my parents for giving me the opportunities and experiences that have made me who I am.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

IR	Infrared Radiation
MWIR	Mid-Wave Infrared Radiation
DBT	Detect-Before-Track
TBD	Track-Before-Detect
2D	2 Dimensional
3D	3 Dimensional
JPDA	Joint Probabilistic Data Association
MHT	Multiple Hypothesis Filter
PMHT	Probabilistic Multiple Hypothesis Filter
NCC	Normalized Cross Correlation
MOSSE	Minimum Output Sum of Squared Error
FFT	Fast Fourier Transform
SVM	Support Vector Machine
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
HMM	Hidden Markov Model
OTA	Overall Tracking Accuracy
OTP	Overall Tracking Precision
ATA	Average Tracking Accuracy

CHAPTER 1

INTRODUCTION

1.1 Motivation and Objective

Object tracking is a fundamental problem in computer vision, employed in a wide range of applications such as surveillance, target tracking in defence, human-computer interaction and medical imaging. Object tracking has been studied for decades with substantial progress but still remains a challenging problem.

In a typical naval warfare scenario, tracking the positions of enemy surface watercraft is highly crucial. Apart from radars, tracking of surface watercrafts is commonly performed on images gathered from an infrared camera on a flying platform. In this thesis, employing object tracking algorithms for tracking surface watercrafts in airborne infrared imagery is studied.

1.2 Problem Definition

1.2.1 Object Tracking

Object tracking can be defined as the estimation of the object state in subsequent frames, given the initial state of the object. State of the object is generally the geometric properties of the object in the image frame; such as the central position, bounding box and silhouette of the object. But the state can also be defined as the 3D position of the object in a real-world reference coordinate system.

If the object appears large enough in the image frame such that it exhibits distinguishing features, its appearance features can be recognized in subsequent frames. Thus, it is required to employ a different approach for tracking small and/or faint objects that has no distinguishing appearance feature. So, in this work, object tracking algorithms are divided into two categories: small object tracking and visual object tracking.

Small target tracking algorithms focus on the problem of measurement origin uncertainty. When the object being tracked is small, only information gathered from infrared sensor is the centroid positions of small object detections. There is no distinguishing appearance information which can be used to discern our object from other objects or false detections. So it is not trivial to determine the origin of small target centroid measurements. The procedure of matching the correct measurement and the object being tracked is called *data association*. Data association algorithms are being studied for decades and there is a substantial amount of work in the field.

Visual target tracking algorithms use a representation of the object's appearance for locating the object in subsequent frames. Appearance representation can be simply the intensity values or some other representation as histogram, silhouette, image gradients or automatically generated feature maps by convolutional neural networks.

1.2.2 Surface Vessel Tracking

In a naval warfare environment, tracking the positions of the enemy water craft is highly critical. This is achieved by mainly using radars. But, radar systems emits electromagnetic waves and this conversely can be used by enemy water craft to determine the position of the radar system. Tracking the targets using infrared imagery is a passive alternative which emits no energy. Tracking the positions of surface vessels is generally performed on imagery acquired by a flying platform.

There are several problems which are specific to tracking surface combatants in infrared images acquired by a flying platform. The most encountered problem is infrared countermeasures used by the target. If a warship detects an incoming flying threat, it releases decoy heat sources into the air to mislead the threat. Most visual target tracking algorithms are robust to this kind of objects in the scene; because most of the time, shape of the decoy differs significantly from the target. However, if the decoy partially overlaps with the target or completely blocks the view of the target, tracking performance critically decreases. This scenario especially occurs when the target uses a system called trainable decoy launcher which actively aims to block the view of the threat by releasing multiple heat decoys to create a big hot cloud in the direction of the incoming threat. Occlusion of the target also occurs when the platform which acquires the infrared image is flying at a low altitude. This means image is acquired with a low pitch angle. This low pitch angle causes ship targets to occlude each other in the case of tracking ships of a fleet. Target tracking algorithms generally detect the occlusion of the target by thresholding its confidence metric and deciding that target being tracked is absent in the current frame. Tracker then commonly stops updating its target appearance model to prevent the model from diverging to a non-target object in the scene.

Modern warships also employ active cooling systems by spraying sea water onto itself in order to reduce the temperatures of the outer body of the ship. This makes ship appear in the infrared images with similar pixel values with the sea background. This similar appearance with the background may cause a tracker to match with a background region resulting in decreased tracking performance or a complete loss of the target.

A fast manoeuvring imaging platform coupled with a manoeuvring target can cause target's appearance on the image change quickly over time. A tracker must adapt its target appearance model quickly or use an appearance model which is invariant under the changing pose of the target.

1.3 Our Approach and Contribution

In this work, performances of state of the art, visual target tracking algorithms are evaluated on synthetic infrared imagery. Small target tracking algorithms are excluded from our tests and it is assumed that a small target tracking procedure is utilized with optimal performance until the starting frame of the visual tracking algorithm. Imagery dataset consists of synthetic mid-wave (MW) infrared imagery captured from a flying platform in several scenarios. These scenarios are typical naval warfare air engagements which typically include a flying platform approaching to a naval surface vessel with different altitudes, angles and speeds under different atmospheric and lighting conditions. These engagements may or may not occur in the presence of other, out of interest surface vessels.

A variety of visual target tracking algorithms are selected to evaluate their performances in our dataset. Algorithms are selected by their popularity in the visual tracking literature. The selected algorithms for this work are Mean Shift Tracking [14], Normalized Cross Correlation [37], Minimum Output Sum of Squared Error Filter [7], Structured Output Tracking [21] and Hierarchical Convolutional Features for Visual Tracking [38].

In order to measure the performances of the selected trackers, several notable visual tracking metrics are investigated. Two tracking metrics are found to be the best suited for our problem. Tracking metrics used in this work are named F-score [35] and Deviation [45].

Selected algorithms are evaluated with selected metrics on our IR surface vessel imagery dataset. Hierarchical Convolutional Features for Visual Tracking algorithm has outperformed the other algorithms. However, this algorithm is designed to track all kinds of objects because it is a general object tracker. Since the main goal of this thesis is to track watercraft, the best scoring algorithm is modified to track only specific types of objects. This modification consists of retraining the underlying feature extracting convolutional neural network. The network is trained on a dataset which consists of IR images of different surface vessels and different background clutter types. An occlusion detection procedure is also employed in the matching procedure to improve the tracking performance in the cases where target is occluded or not visible by any other cause for a limited amount of time. It was seen that these modifications improved the performance of the tracker in our dataset, which consists of IR images of combat ships.

1.4 Organization of the Thesis

This thesis consists of 5 chapters. Chapter 1 introduces the motivation for this thesis, defines the problem of focus and describes our approach and contribution to the problem.

Chapter 2 contains literature survey for both small and visual target tracking problems. This chapter describes significant works carried through in the field.

Chapter 3 introduces theoretical background for target tracking algorithms and describes notable target tracking algorithms in detail. This chapter also lays the theoretical groundwork for this thesis.

Chapter 4 describes how target tracking algorithm performances are evaluated with our dataset. Performance evaluation metric and out dataset is described in detail. This chapter also presents experimental results of our evaluation. Performance metric scores and survival curves for each tracking algorithm is presented and the results are discussed in detail.

Finally, chapter 5 concludes the thesis with discussions about experimental results and inferences made from the results. This chapter also mentions possible future work on the problem.

CHAPTER 2

LITERATURE SURVEY

2.1 Small Object Tracking

Since very small targets have no distinguishing features other than appearing as small points in the images; tracking methods such as kernel tracking and silhouette tracking are not suitable for small target tracking. Small target tracking methods can be categorized in two main topics: detect-before-track (DBT) and track-before-detect (TBD) algorithms.

Detect-before-track algorithms first declare target detections in each step. Target detection is performed by a small target detection algorithm and detection results are used to estimate target trajectory [52]. If there is one small target to be tracked, Kalman filter and particle filter DBT algorithms are mainly used. An iterated extended Kalman filter is used in [10] to estimate object motion from noisy images. [4] uses Kalman filter to estimate object location and speed in 3D coordinates in stereo images. An extended Kalman filter is used in [44] to estimate 3D trajectory from 2D image motion. In the case of non-Gaussian distribution of target motion state variables, Kalman filter gives poor estimation results. This problem can be overcome by particle filters which do not assume normally distributed state variables. A randomly generated set (particles) is used for modelling non-Gaussian distribution of target motion parameters in clutter [27]. A real-time object recognition system using particle filter is implemented in [13].

When tracking multiple small targets, problem of corresponding detections and tracked objects arises. The correspondence problem needs to be solved before Kalman or particle filters can be applied. Joint Probabilistic Data Association (JPDA) and Multiple

Hypothesis Tracking (MHT) are two widely used techniques for solving the correspondence problem [50]. JPDA is defined in [20], and used for tracking multiple targets in a cluttered environment. Major limitation of JPDA algorithm is that it cannot handle changing number of targets. In [11], a method to track variable number of targets using JPDA is proposed. JPDA produces erroneous results when a target exits field of view or a new target enters field of view. Multiple hypothesis algorithm does not have this drawback [50]. [43] defines MHT as a multiple target tracking algorithm with track initiation and deletion capabilities. MHT considers all possible measurement-to-target associations called hypotheses in each iteration and generates new hypotheses from existing hypothesis recursively. This branching approach generates a tree representing probabilities for all possible associations from the very first measurement [43]. Number of hypotheses grows exponentially with new measurements. Since it creates excessive amount of workload for the algorithm platform, [15] chooses k-best hypotheses and discards other inferior hypotheses. Standard MHT approach uses a Kalman filter for state estimation of the tracks, [25] uses a particle filter as the state estimator of the multiple hypothesis tracking. In [48], hypotheses is not represented as a list or a tree, instead hypotheses are defined as association probabilities. This approach is called Probabilistic Multiple Hypothesis Tracking (PMHT).

Track-before-detect algorithms collect the energy of target candidates before declaring some of the candidates as targets. Measurement data without any thresholding is used as an input to tracker. If any track exceeds certain target likelihood, it is declared as a target. The main problem is that sensor image is a highly non-linear function of the target state [17]. A method to solve this problem is to discretize the state space. Discrete target state space makes it possible to use Viterbi algorithm, [1] uses Viterbi algorithm with dynamic programming approach to track and detect small targets in a heavily cluttered environment. [42] uses 3-D matched filter to detect small objects. A set of filters with different velocity assumptions for the target and the image sequence is multiplied in 3D frequency domain and the result is inverse transformed.

2.2 Visual Object Tracking

If target size is large enough such that it has unique features that can be tracked, kernel based and silhouette based trackers are suitable for tracking such targets in image sequences. Kernel based tracking methods can be categorized in two main topics: template-based tracking and density-based tracking.

Template matching is a common approach for template-based tracking. An intensity based template representing the target region is searched in every new frame to find the new location of the object in the image [50]. New location of the target is computed by a similarity measure such as sum of squared distances and cross-correlation. There are several problems for standard template matching approach. Illumination changes on the object produces low similarity values with the template and makes it possible for template to match with a non-target region. [5] uses intensity gradients, which are invariant under illumination changes, for template matching. Another approach for handling changing illumination is normalizing the filter and image under the filter window. [9] uses Normalized Cross Correlation (NCC) as the similarity measure for template matching. Another problem for template matching is changing appearance of the target due to scaling and/or rotation. In order to address this problem, a dynamically evolving template is used. Template is updated with the weighted sum of the old template and the intensity values in the matched region in new frame [39]. [7] introduces Minimum Output Sum of Squared Error (MOSSE) filter which finds a template such that it minimizes the sum of squared error between actual correlation output and desired correlation output. Desired correlation output is a strong peak at the location of the target and zero intensity elsewhere. This technique allows template to adapt to changing appearance of the target. Evolving template approach can handle scale changes as long as object region is smaller than the template window. If target becomes larger than the window, changing the size of the template becomes a necessity. In order to find a suitable windows size for the target, some tracking algorithms aim to estimate target scale. A scale dimension is defined on the image in addition to horizontal and vertical spatial dimensions. Tracker searches template in translation and scale. After finding the correct scale, tracker updates the template size according to target scale. [34] extends MOSSE by introducing separate filters for scale and translation for robust estimation of the scale. This resulted a tracker which outperforms other state-of-the-art tracking methods [16]. Standard template matching approach is also slow in terms of computation speed because it searches the whole image for a match with a brute search. One solution to address this problem is only searching for a match in the neighbourhood of the previous match point. Another solution is computing the correlation in frequency domain, [37] introduces a Fast Fourier Transform (FFT) based NCC method to track non-moving objects in order to estimate camera motion. [24] observes that a discriminative template takes a circulant structure when trained with large number of samples. The circulant structure is exploited to achieve very fast update of the the filter and detection of the object in the new frame.

Another kernel based tracking method is density-based tracking. Density based trackers uses probability density of the object region as the feature for tracking. Probability density of the object region is usually represented by intensity histograms. [14] uses weighted histogram of the circular target region to track objects. Instead of searching whole image, authors use mean-shift approach to find the best histogram match. Mean-shift procedure is initialized from the target location found in the previous frame, circular target window then iteratively moves in the direction of the mean-shift vector to the new location of the target. One problem of mean-shift tracking is that filter size is constant. [8] uses an elliptically shaped filter and adapts filter size and orientation to the new appearance of the object. [28] uses three probabilistic component mixture as a model for the object region. The three components defined as static features, transient features and noise component. The static features represents the most reliable features of the target; transient features models changing appearance of the object; noise part is for modelling outlier pixels. Another density based approach is presented in [22]. Target region is represented by a grid of multiple local histograms and target is search around a neighbourhood by the distance between template histograms and the region of interest.

Classification based trackers are emerging recently. These trackers employs classification algorithms to classify each region in the new frame as target or background. Tracker also trains classification algorithm during tracking. [21] presents an adaptive tracker based on structured support vector machine (SVM) with online learning capabilities. Authors also introduces a control mechanism to keep the number of support vectors at a suitable bound. [40] uses convolutional neural networks (CNN) for tracking. A number of layers called domain-specific layers are trained with separate annotated videos. Then, trained domain-specific layers are combined to obtain a shared layer with generic target representation. The resulting tracker is very accurate and robust but slow in execution time, because online training of a CNN is a costly operation [33]. In [49], an offline trained stacked denoising autoencoder is fined tuned online to adapt to appearance changes of the target. TLD (Tracking-Learning-Detection) [29] divides the localization of the object into two separate parts: tracking and detection. A classifier detects and corrects the errors of tracking and detection while learning the appearance of the target. In [23], authors feed the current and previous frames to two separate convolutional neural networks. A third network is trained on the outputs of these two networks and the object translation between the frames. No online training is performed during tracking.

Silhouette based trackers are usable for tracking objects that has non-rigid changing complex shape or their silhouette change because of 3D rotation. Silhouette based trackers use object contour as the representation of the object being tracked. [26] proposes a contour tracking algorithm with a novel approach to conditional density propagation. Object shape and position model are represented with factored samples. [51] presents an active contour based tracker which is robust to occlusions. Shape priors are used to recover the shape of the occluded object. A hidden Markov model (HMM) based silhouette tracking algorithm is presented in [12] . Authors also employs joint probabilistic data association filter to establish a region smoothness constraint in addition to contour smoothness constraint.

CHAPTER 3

THEORETICAL BACKGROUND

Target tracking in a machine vision scope can be described as the estimation of the state of an object in a frame using the information about the object from previous frames. For most applications, state of the object is simply the centroid pixel location of the object and in most cases bounding box of the object in the frame. In applications which the outer geometry of the scene is concerned about, state is generally 3D position of the object relative to real world or the imaging platform.

3.1 Small Target Tracking

Small Target Tracking is a case of target tracking which the only measurement about the object is the centroid pixel location. In this case there is no discriminating features about the object is available since the object appears only as a small point in the image. Conditions which create this appearance is dependent on the size of the object, distance to the object, optical and radiometric characteristics of the camera. But from a machine vision perspective, the conditions which create the small appearance is generally irrelevant and simply the pixel area of the object in the image frame is considered to treat the object as a small target. In a naval combat environment tracking of watercrafts generally starts in a distance where targets suit the definition of small target. Since in this case camera only acts as a position detector, small target tracking algorithms can work with position data provided with a sensor type different from a camera such as radars or sonars.

In order to overcome the problem of discriminating the objects from each other, small target tracking algorithms employ state estimators and data association algorithms.

Data association algorithms compare the position measurements from the acquired image and state predictions produced by the state estimator to differentiate objects from each other. In following sections the most distinguished state estimator, Kalman filter, and two popular data association techniques are explained to give a perspective about the small target tracking methodology.

3.1.1 Kalman Filter

Kalman filter [30] is an estimator which estimates the state of a linear system. It uses measurements that are generated as a linear function of the system state but corrupted by additive Gaussian noise. Kalman filter employs two procedures called time update(prediction) and measurement update(correction). Let x_k be the state of the system at time step k, then state transition at each time step can be written as a linear combination of the previous state, a control signal and noise:

$$x_k = Ax_{k-1} + Bu_k + w_k \tag{3.1}$$

where A is a matrix called state transition matrix, u_k is a control signal assuming the system is controlled by another system, B is a matrix representing the control model and w_k is the process noise. Measurements from the system can be modelled as a linear transform of the state with additive Gaussian noise:

$$z_k = Hx_k + v_k \tag{3.2}$$

where z_k is the measurement, H is the measurement model and v_k is the measurement noise. A, B and H matrices of the system are constant and must be known or correctly modelled for Kalman filter to work correctly.

3.1.1.1 Time Update (Prediction)

Kalman filter makes a prediction about the state of the system in next iteration. The prediction is simply made by using the state transition model:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{3.3}$$

where $\hat{x}_k^-$ is the state prediction or the state prior. Since it is an estimator, Kalman filter has a state estimation covariance matrix denoted as P_k. Prediction step also makes a prediction about what estimation covariance will be at next iteration, in other words it projects ahead the error covariance for one step:

$$P_k^- = AP_{k-1}A^T + Q \tag{3.4}$$

where Q is the covariance matrix of the process noise w_k.

3.1.1.2 Measurement Update (Correction)

In correction equations, new measurement about the system is used to update previously predicted state estimate. Firstly, Kalman gain, denoted as K_k is calculated. Kalman gain is a weighting factor that determines how much the new measurement affects the new estimate of the state. Kalman gain is calculated by using the state estimation covariance matrix P_k and measurement covariance matrix R:

$$K_k = P_k^- H^T(HP_k^- H^T + R)^{-1} \tag{3.5}$$

Note that R is a constant user defined value, so one must correctly define how noisy the measurements would be. State prediction is updated with a value called the measurement prediction error or the residual.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{3.6}$$

$H\hat{x}_k^-$ is what would the measurement be if the state prediction is correct. Therefore $(z_k - H\hat{x}_k^-)$ is the difference between the measurement z_k and predicted measurement $H\hat{x}_k^-$, so it is the measurement residual. State estimate is corrected by the measurement residual weighted by Kalman gain. Correction step also corrects the state estimation covariance prediction P_k^- using calculated Kalman gain:

$$P_k = (I - K_k H)P_k^- \tag{3.7}$$

Prediction and correction procedures run at every iteration respectively. Outputs of every iteration are used for inputs to next iteration.

3.1.2 Joint Probabilistic Data Association Filter

In a small target tracking system, Joint Probabilistic Data Association Filter (JPDAF)[20] is an algorithm that matches measurements to existing target tracks. Following the measurements, JPDAF calculates the probabilities of all possible joint measurement-to-track associations. A joint association is a decision representing a joint event, which consists of all measurement events in a time step. JPDAF defines a joint event with following:

$$\boldsymbol{\theta} = \bigcap_{j=1}^{m_k} \theta j t_j \tag{3.8}$$

where

$$\begin{aligned} \theta jt &\triangleq \{\text{measurement } j \text{ originated from target } t\}, \\ j &= 1, ..., m_k; \quad t = 0, 1, ..., T \end{aligned} \tag{3.9}$$

T is the number of targets being tracked and m_k is the number of measurements in time step k (latest time step). Instead of considering all joint events, JPDAF uses a validation matrix to eliminate associations with negligible probability in order to reduce computational complexity. If a measurement j is outside of the validation gate of target t, association probability of the event θ_{jt} is considered negligible. JPDAF defines a validation matrix as follows:

$$\boldsymbol{\Omega} \triangleq [\omega_{jt}], \qquad j = 1, ..., m_k; \quad t = 0, 1, ..., T \tag{3.10}$$

where ω_{jt} is a binary value that indicates if measurement j is in the validation gate of target t. Validation matrix is used for generating different joint event permutations. A joint event $\boldsymbol{\theta}$ is represented in the matrix form similar to validation matrix.

$$\hat{\boldsymbol{\Omega}}(\boldsymbol{\theta}) = [\hat{\omega}_{jt}(\boldsymbol{\theta})] \tag{3.11}$$

$\hat{\omega}_{jt}(\boldsymbol{\theta})$ is 1 if θ_{jt} is an association event in $\boldsymbol{\theta}$. $\hat{\boldsymbol{\Omega}}(\boldsymbol{\theta})$ has one extra column for representing false alarms (i.e., measurements which are not originated from a target). A joint event is called *feasible* if a measurement can only be originated from one source, i.e.,

$$\sum_{t=0}^{T} \hat{\omega}_{jt}(\boldsymbol{\theta}) = 1, \qquad j = 1, ..., m_k \tag{3.12}$$

and no more than one measurement can originate from a target, i.e.,

$$\delta_t(\boldsymbol{\theta}) \triangleq \sum_{j=1}^{m_k} \hat{\omega}_{jt}(\boldsymbol{\theta}) \leq 1, \qquad t = 1, ..., T \tag{3.13}$$

$\delta_t(\boldsymbol{\theta})$ is called *target detection indicator*. It indicates that target t is associated with a measurement in the joint event $\boldsymbol{\theta}$. For the convenience of the main equation of JPDAF, a binary *measurement association indicator* $\tau_j(\boldsymbol{\theta})$ is also defined. It indicates that measurement j is associated with a target.

$$\tau_j(\boldsymbol{\theta}) \triangleq \sum_{t=1}^{T} \hat{\omega}_{jt}(\boldsymbol{\theta}), \qquad j = 1, ..., m_k \tag{3.14}$$

Using $\tau_j(\boldsymbol{\theta})$, number of unassociated measurements can be calculated:

$$\phi(\boldsymbol{\theta}) = \sum_{j=1}^{m_k} [1 - \tau_j(\boldsymbol{\theta})] \tag{3.15}$$

JPDAF calculates the probability of a feasible joint event with as follows:

$$P\{\boldsymbol{\theta}|Z\} = \frac{1}{c}\frac{\phi!}{V^{\phi}} \prod_{j=1}^{m_k} \Big[N_{t_j}[\mathbf{z}_j]\Big]^{\tau_j} \prod_{t=1}^{T} (P_D^t)^{\delta_t}(1 - P_D^t)^{1-\delta_t} \tag{3.16}$$

where Z is the measurement set, $\mathbf{z}_j$ is an individual measurement, c is a normalization constant, V is the surveillance region volume, N_{t_j} is the multivariate normal distribution function generated with the covariance matrix produced by the state estimator of target t_j. P_D^t is the probability of detection of target t. After calculating the probabilities of every feasible joint event, marginal probabilities are calculated as follows:

$$\begin{aligned} \beta_{jt} &\triangleq P\{\theta jt|Z\} \\ &= \sum_{\boldsymbol{\theta}:\theta jt \in \boldsymbol{\theta}} P\{\boldsymbol{\theta}|Z\} \end{aligned} \tag{3.17}$$

State estimation of each target's Kalman filter is performed by using a combined innovation, which is calculated as follows:

$$v_t = \sum_{j=1}^{m_k} \beta_{jt} v_{jt} \tag{3.18}$$

3.1.3 Multiple Hypothesis Filter

Multiple hypothesis filter [43] is an algorithm to resolve measurement-to-track association problem in multi-target tracking systems. Unlike joint probabilistic data association filter (JPDAF) which only considers latest set of measurements and already established tracks; multiple hypothesis filter generates and holds hypotheses about all possible association decisions from the beginning, including initiation of a new track. In each iteration, association decision is chosen as the hypothesis with the highest probability while other hypotheses are retained and improved. This prevents an incorrect association to cause a complete mix up of tracks.

Let $\mathbf{\Omega}^k$ be the set of all association hypotheses up to time step k. $\mathbf{\Omega}^k$ consists of joint cumulative events. A joint cumulative event $\Theta^{k,l}$ at time step k can be defines as follows:

$$\Theta^{k,l} = \{\Theta^{k-1,s}, \boldsymbol{\theta}(k)\} \tag{3.19}$$

where $\boldsymbol{\theta}(k)$ is a joint event in time k which consists of τ number of measurements originated from an existing track, v number of measurements originated from new targets and ϕ number of measurements that are false alarms. Following indicators are defined for latest set of measurements $\boldsymbol{z}_i(k)$, $i = 1...m_k$

$$\tau_i = \tau_i[\boldsymbol{\theta}(k)] \triangleq \begin{cases} 1, & \text{if } \boldsymbol{z}_i(k) \text{ originated from an existing track} \\ 0, & \text{otherwise} \end{cases} \tag{3.20}$$

$$v_i = v_i[\boldsymbol{\theta}(k)] \triangleq \begin{cases} 1, & \text{if } \boldsymbol{z}_i(k) \text{ originated from a new target} \\ 0, & \text{otherwise} \end{cases} \tag{3.21}$$

$$\delta_t = \delta_t[\boldsymbol{\theta}(k)] \triangleq \begin{cases} 1, & \text{if track } t \text{ is detected at time step } k \\ 0, & \text{otherwise} \end{cases} \tag{3.22}$$

Joint cumulative events in $\mathbf{\Omega}^{k-1}$ is augmented with feasible joint association events in time srep k to create $\mathbf{\Omega}^k$. Conditional probability of each cumulative event is calculated as follows:

$$\begin{aligned} P\{\Theta^{k,l}|Z^k\} = \frac{1}{c}\frac{\phi! v!}{m_k V^{\phi v}}\mu_F(\phi)\mu_N(v)\prod_{i=1}^{m_k}\Big[N_{t_i}[\mathbf{z}_i(k)]\Big]^{\tau_i} \\ \times \prod_{t=1}^{T}(P_D^t)^{\delta_t}(1-P_D^t)^{1-\delta_t}P\{\Theta^{k-1,s}|Z^{k-1}\} \end{aligned} \tag{3.23}$$

where Z^k is the set of all measurements up to time step k, $\mathbf{z}_i(k)$ is an individual measurement, c is a normalization constant, V is the surveillance region volume, N_{t_j} is the multivariate normal distribution function generated with the covariance matrix produced by the state estimator of target t_i. P_D^t is the probability of detection of target t. $\mu_F(\phi)$ and $\mu_N(v)$ are the probability mass functions of number of false alarms and number of new targets. After calculating the probabilities of all joint cumulative events in $\mathbf{\Omega}^k$, MHF chooses joint cumulative event with the highest probability as the association decision.

Number of hypotheses grow exponentially as new measurements arrive. MHT performs hypothesis reduction techniques to keep the number of hypotheses at a reasonable level. Hypotheses with negligible probabilities are eliminated and hypotheses which have similar associations are merged.

3.2 Visual Target Tracking

Visual tracking can be defined as continuously estimating the new location of the object in the new acquired image using the previous information about the appearance of the target. Most of the visual target tracking algorithms can be divided into two main parts such as target appearance model and target matching. Appearance model is a representation of the past information gathered about the target. In this model, appearance of the target can be represented by an intensity template, intensity histogram or feature vectors [47]. Target matching is the stage where the tracking algorithm tries to find the most similar region in the new frame to its target appearance model in order to locate the target. While it is also possible to take advantage of target state estimators, as in the case of small target trackers, to help with the matching state; our work concentrates on tracking using appearance information only. In the following part of this section, several distinguished visual tracking algorithms are explained in detail.

3.2.1 Mean Shift Tracking

Mean-shift [14] is a kernel based object tracking algorithm. Mean-shift uses weighted histogram of the target region as the kernel. Object is localized by histogram similar-

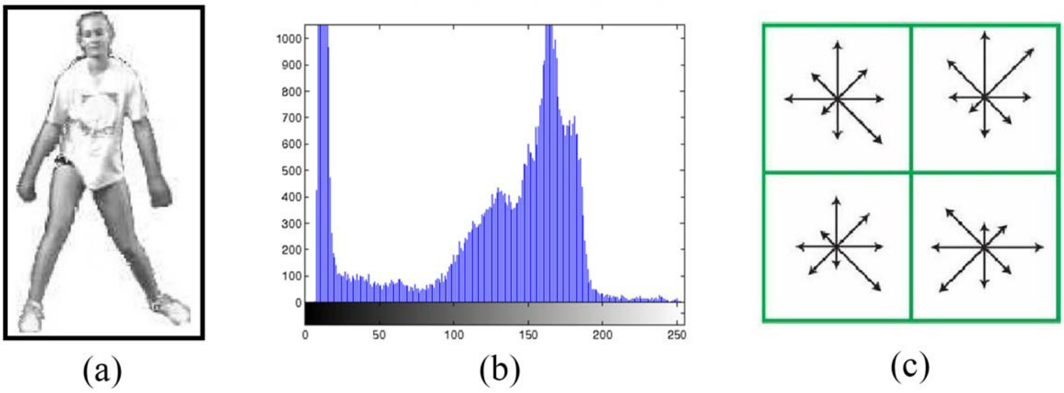


Figure 3.1: Popular appearance models [47]. (a) Intensity template [50], (b) intensity histogram and (c) feature vectors.

ities between the kernel and the object region in the new frame. Instead of searching the most similar histogram exhaustively, kernel window moves from the previous location to the new location of the object by a mean-shift procedure. Mean-shift employs Bhattacharyya coefficient to define a distance metric between two histograms. Bhattacharyya coefficient has the following form in terms of densities:

$$\rho(\boldsymbol{y}) \equiv \rho[p(\boldsymbol{y}), q] = \int \sqrt{p_{\boldsymbol{z}}(\boldsymbol{y}) q_{\boldsymbol{z}}} d\boldsymbol{z} \tag{3.24}$$

Since Mean-shift uses weighted histograms as density estimates, Bhattacharyya coefficient takes the following form:

$$\hat{\rho}(\boldsymbol{y}) \equiv \rho[\hat{\mathbf{p}}(\boldsymbol{y}), \hat{\mathbf{q}}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\boldsymbol{y}) \hat{q}_u} \tag{3.25}$$

where $\hat{q}_u$ is the target histogram, $\hat{p}_u(\boldsymbol{y})$ is the histogram of the region centered at discrete pixel location **y** and m is the number of histogram bins. Based on Bhattacharyya coefficient, a distance metric between two histograms is defined:

$$d(\boldsymbol{y}) = \sqrt{1 - \rho[\hat{\mathbf{p}}(\boldsymbol{y}), \hat{\mathbf{q}}]} \tag{3.26}$$

3.2.1.1 Weighted histogram computation

Pixel locations of the region centered around $\boldsymbol{y}$ are denoted by $\{\boldsymbol{x}_i\}_{i=1...n}$. Let $b : B^2 \rightarrow \{1...m\}$ be a function which maps $\boldsymbol{x}_i$ to the index $b(\boldsymbol{x}_i)$ of the histogram bin corresponding to intensity value at the pixel location $\boldsymbol{x}_i$. The weighted histogram can be written as

$$\hat{q}_u = C \sum_{i=1}^{n} k\left(\left\|\frac{\boldsymbol{y} - \boldsymbol{x}_i}{h}\right\|^2\right) \delta[b(\boldsymbol{x}_i) - u] \tag{3.27}$$

where C is a normalization constant, h is the scale of the region, δ is the Kronecker delta function, $k : [0, \infty) \rightarrow R$ is a function which assigns smaller weights to the pixel locations farther away from the center.

3.2.1.2 Distance Minimization

Search procedure to find the new location y of the target starts from the location $\boldsymbol{y}_0$ of the target in the previous frame. Minimization of the distance metric is equivalent to maximization of Bhattacharyya coefficient. Taylor expansion of $\rho[p(\boldsymbol{y}), q]$ yields:

$$\rho[p(\boldsymbol{y}), q] \approx \frac{1}{2} \sum_{u=1}^{m} \sqrt{\hat{p}_u(\boldsymbol{y}_0)\hat{q}_u} + \frac{1}{2} \sum_{u=1}^{m} \hat{p}_u(\boldsymbol{y}) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\boldsymbol{y}_0)}} \tag{3.28}$$

Since first term is independent of y, second term has to be maximized. Introducing (4) to second term, we get

$$\rho[p(\boldsymbol{y}), q] \approx \frac{1}{2} \sum_{u=1}^{m} \sqrt{\hat{p}_u(\boldsymbol{y}_0)\hat{q}_u} + \frac{C}{2} \sum_{u=1}^{n_h} w_i k\left(\left\|\frac{\boldsymbol{y} - \boldsymbol{x}_i}{h}\right\|^2\right) \tag{3.29}$$

where

$$w_i = \sum_{i=1}^{m} \delta[b(\boldsymbol{x}_i) - u] \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\boldsymbol{y}_0)}} \tag{3.30}$$

Maximization is achieved by Mean-shift procedure. Given the distribution $\hat{q}_u$ and the previous location y_0 of the target:

1. Compute the weights $\{w_i\}_{i=1\ldots n_h}$ according to (7).

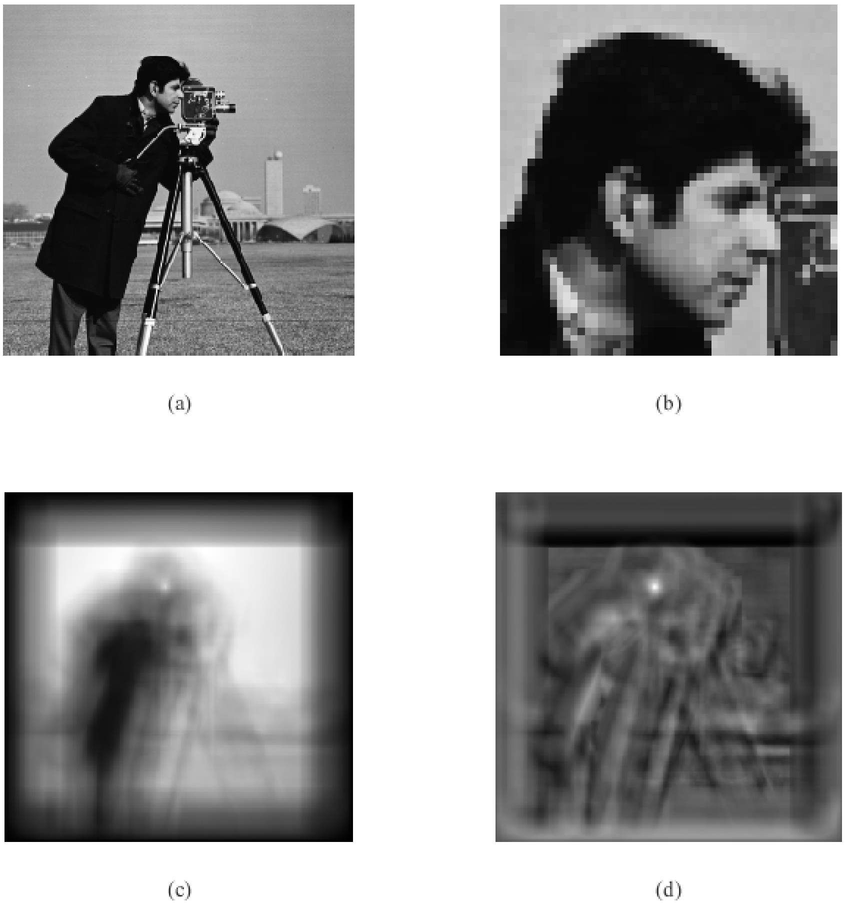
2. Calculate the new location of the target

$$\boldsymbol{y}_1 = \frac{\sum_{i=1}^{n_h} \boldsymbol{x}_i w_i g\left(\left\| \frac{\boldsymbol{y}_0 - \boldsymbol{x}_i}{h} \right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\| \frac{\boldsymbol{y}_0 - \boldsymbol{x}_i}{h} \right\|^2\right)} \tag{3.31}$$

3. If $\|\boldsymbol{y}_1 - \boldsymbol{y}_0\| < \epsilon$ Stop.
   Otherwise Set $\boldsymbol{y}_0 \leftarrow \boldsymbol{y}_1$ and go to Step 1.

3.2.2 Normalized Cross-Correlation

Normalized Cross-Correlation (NCC) Tracking [37] is a template matching based tracking algorithm which uses NCC to find the closes match to template in the new frame. NCC differs from standard correlation operation by preprocessing the inputs to have zero mean and unit variance. This property of NCC prevents the template to match with high intensity, non-target regions. Figure 3.2 shows the result of the same template with conventional cross-correlation and NCC. It can be seen that NCC allows accurate template matching as it only gives high correlation output on the location of the actual template.



(a) (b)

(c) (d)

Figure 3.2: Conventional cross-correlation output (c) and NCC output (d) using the image (a) and template (b).

NCC operation between two images that have the same size defined as follows:

$$\mathcal{N}_{\times}(\boldsymbol{A},\boldsymbol{B}) = \frac{\sum_i(\boldsymbol{A}(p_i)-\bar{\boldsymbol{A}})(\boldsymbol{B}(p_i)-\bar{\boldsymbol{B}})}{\sqrt{\sum_i(\boldsymbol{A}(p_i)-\bar{\boldsymbol{A}})^2}\sqrt{\sum_i(\boldsymbol{B}(p_i)-\bar{\boldsymbol{B}})^2}} \tag{3.32}$$

NCC operation is performed on the whole image by moving the template window over the image. Since it is necessary to calculate the mean and variance of every region which template moves over, NCC is a costly operation to perform. In order to address this problem, [37] employs integral images to calculate NCC by Fast Fourier Transform.

After the NCC matching operation and finding the maximum of the result, region around the newly found target location is blended with the template to generate the new filter for the next image.

$$T_{k+1} = \alpha I(\boldsymbol{x}) + (1 - \alpha)T_k \tag{3.33}$$

where T is template, k is time step, x is the detected target location and α is the blending ratio.

3.2.3 Minimum Output Sum of Squared Error Filter

Minimum Output Sum of Squared Error (MOSSE) Filter [7] is a template based, adaptive correlation filter for tracking objects in images. Correlation operation in the Fourier domain can be written as follows:

$$G = F \odot H^* \tag{3.34}$$

where F is the current frame image and H is the filter. The location of the object in the current frame is determined by finding the maximum of the correlation output g, which is the inverse Fourier transform of G. MOSSE defines a desired correlation output which is a strong peak at the centroid location of the object being tracked. In order to find a filter that produces desired correlation output, MOSSE minimizes the distance between the actual correlation output and the desired correlation output.

$$\min_{H^*} |F \odot H^* - G| \tag{3.35}$$

A closed form expression for the MOSSE filter is derived by solving for H^*:

$$H^* = \frac{G \odot F^*}{F \odot F^*} \tag{3.36}$$

A filter calculated with this approach fits exactly to current frame but will often fail for locating the object in the next frame. To make the filter more general, an averaging approach is used. Numerator and denominator of the filter extracted from the current frame are named A_i and B_i and averaged separately with the numerator and denominator values of the previous frame (A_{i-1}, B_{i-1}):

$$\begin{aligned} A_i &= \eta G_i \odot F_i^* + (1-\eta)A_{i-1} \\ B_i &= \eta F_i \odot F_i^* + (1-\eta)B_{i-1} \\ H_i^* &= \frac{A_i}{B_i} \end{aligned} \tag{3.37}$$

where η is the averaging ratio. Large η puts more emphasis on the current frame.

Initialization of the filter is performed on the initial image by sampling N different regions around the target. The locations of the samples are determined by slightly perturbing the location of the original target location. Initial A and B are calculated as follows:

$$\begin{aligned} A &= \frac{1}{N}\sum_i G_i \odot F_i^* \\ B &= \frac{1}{N}\sum_i F_i \odot F_i^* \end{aligned} \tag{3.38}$$

where F_i is the Fourier transform of the i th perturbed region sample and G_i is the corresponding desired output for the region.

3.2.3.1 Occlusion Detection

One of the main features of MOSSE tracking algorithm is being able to detect if the object of interest is not visible in the current frame. This feature aims to stop appearance template update to prevent filter from being updated by non-object regions. Object visibility is determined by measuring the peak strength of the maximum value in the correlation output. If the peak is not strong enough, it is decided that the object is not present in the current frame. Peak strength is measured by a metric called Peak to Sidelobe Ratio (PSR). PSR measures the peak strength by dividing the correlation output g into two seperate regions. First region is the peak which is the maximum value of g and second region is the sidelobe which is the rest of the image excluding a 11×11 region around the maximum value. PSR then is calculated as follows:

$$PSR = \frac{g_{max} - \mu_{sl}}{\sigma_{sl}} \tag{3.39}$$

where g_{max} is the maximum value of g; μ_{sl} and σ_{sl} are the mean value and standard deviation of the sidelobe.

3.2.4 Struck: Structured Output Tracking with Kernels

Struck [21] is an object tracker which uses *structured output SVM* [6] for localization of the object. Traditional SVM based trackers use a sliding window technique to classify each region of the image. Classification scores are used to determine the position of the object in the new frame. Instead of using a classification function which maps image samples to labels of classes, structured output SVM used in Struck employs a classification function which maps image samples to euclidean location transforms in the image. It eliminates the problem of incorrect labelling of the samples by directly working on the image instead of heuristically generating binary labelled samples.

A prediction function $f : X \to Y$ is proposed to estimate the object transform between two frames, which Y is the space of all transformations. A discriminant function $F : X \times Y \to R$, which is introduced in structured SVM framework, is used to calculate prediction function as follows:

$$\mathbf{y}_t = f(\mathbf{x}_t^{\mathbf{p}_{t-1}}) = \underset{\mathbf{y} \in Y}{\operatorname{argmax}} \, F(\mathbf{x}_t^{\mathbf{p}_{t-1}}, \mathbf{y}) \tag{3.40}$$

A budget technique is also employed to limit the number of support vectors for real-time execution of the tracker.

3.2.5 Hierarchical Convolutional Features for Visual Tracking

Artificial Neural Networks (ANN) are a set of classifiers which has been inspired by biological processes. An ANN mimics how a biological brain works by employing interconnected atomic processing elements called artificial neurons. Each neuron has a set of parameters such as input weights and activation rules. These parameters are optimized in a process called training to make the neural network to perform a certain classification task.

Convolutional Neural Networks (CNN) are a subset of ANN's which specific constraints are defined on neuron connections and input weights to make each layer of the network to correspond to a convolution operation. Convolutional layers are generally followed by unconstrained neural network layers to complete a classification task. CNN's are capable of automatically learning distinctive features of the training data for improved classification performance. CNN's are shown to be an effective tool for image classification applications [36].

It is possible to employ CNN's for object tracking [40, 33, 49, 29, 23]. CNN's have the tools to accommodate the two main parts of the tracking process: appearance model and target matching. Hierarchical convolutional features of the CNN's provides an elaborate appearance representation. On the other hand, target matching is performed by classifying each region in the current image to find the region containing the target. This classification task is performed by methods such as fully connected neural networks, support vector machines (SVM) and correlation filters. Training a reliable neural network or SVM classifier requires a large amount of samples around the target region. Correlation filters eliminates the need of sampling around the target [38]. Correlation filters are also preferred in this work by their computational speed in embedded systems.

In [38] it is observed that shallow layers of a CNN encode more precise position information than the deep layers, whereas deep layers encode semantic information better than the shallow layers. From a target tracking perspective, tracking an object using features from shallow layers offers precise localization of the target while tracking an object using deep layers is more robust to appearance changes.

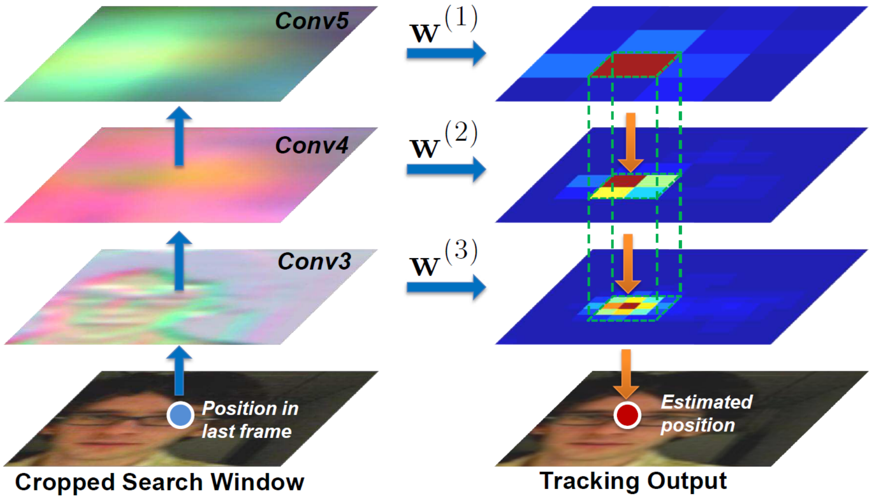


Figure 3.3: Hierarchical location estimation [38]

Authors uses a pre-trained general object classification CNN to extract both deep and shallow convolutional features of an image frame. These features are then used for tracking the object where an adaptive correlation filter bank, similar to MOSSE, is used for each layer to locate the target. Location results are combined in a hierarchical manner to generate a single location for the object in the current image. Figure 3.3 shows the location estimation procedure.

Correlation filter for d-th channel ($d \in \{1, ..., D\}$) of a layer is calculated as follows:

$$H_d^* = \frac{G \odot F_d^*}{\sum_{i=1}^{D} F_i \odot F_i^*} \tag{3.41}$$

where G is the desired correlation result which is a strong peak at the location of the object and F_d is the d-th channel of the feature vector of a layer for the current frame. The correlation result for the l-th layer of the network is calculated as follows:

$$f_l = \mathscr{F}^{-1}(\sum_{d=1}^{D} H_d \odot F_d^*) \tag{3.42}$$

where $\mathscr{F}^{-1}$ is the inverse FFT transform. Target localization for the deepest layer is performed by finding the maximum of the correlation result f_l of size $M \times N$.

$$(\hat{m}, \hat{n}) = \underset{m,n}{\operatorname{argmax}} f_l(m, n) \tag{3.43}$$

For earlier layers, target localization is performed as follows:

$$\begin{aligned} &\underset{m,n}{\operatorname{argmax}} f_l(m, n) + \gamma f_{l+1}(m, n) \\ &\quad s.t. |m - \hat{m}| + |n - \hat{n}| \leq r. \end{aligned} \tag{3.44}$$

Filter update for each layer is performed as follows:

$$\begin{aligned} A_d^t &= \eta G^t \odot (F_d^t)^* + (1 - \eta) A_d^{t-1} \\ B_d^t &= \eta F_d^t \odot (F_d^t)^* + (1 - \eta) B_d^{t-1} \\ (H_d^t)^* &= \frac{A_d^t}{B_d^t} \end{aligned} \tag{3.45}$$

3.3 Proposed Algorithm

The CNN based tracking algorithm described in 3.2.5 uses a pre-trained CNN for feature extraction. The CNN used by authors is VGG-Net-19 [46] trained on ImageNet [18]. Since the method is developed for general purpose object tracking, a CNN trained on general objects suits the needs for this application. In our problem, feature extraction for specific object types is needed. In this work, a transfer learning method which retrains later layers of the VGG-Net-19 employed in [38] is used. The motivation is to extract features which are better suited for our problem and thus increase the tracking performance of the visual object tracker.

3.3.1 Transfer Learning

Transfer learning in a general sense refers to utilizing information gathered in a problem solving procedure to solve another problem. In CNN applications, a pre-trained CNN can be adapted to solve another but related classification problem. Insufficient

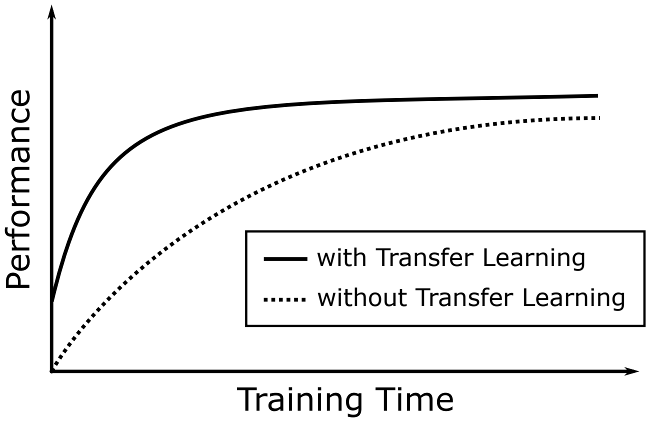


Figure 3.4: Transfer learning training performance comparison

training data is a general problem for CNN training. While using the same training dataset, transfer learning offers faster training times and better overall performance in comparison to initializing a training procedure with random weights. Figure 3.4 compares the training performances of training a CNN with transfer learning and without transfer learning. It is shown that training with transfer learning has better starting performance, better performance slope and better performance asymptote.

Employing transfer learning in CNN training is mainly performed by two methods. First method only replaces the fully-connected layers of the pre-trained network and train these fully-connected layers with new dataset. In this method, convolutional layers are unchanged and used as feature extractors for the new classification problem. Second method allows training algorithm to retrain convolutional layer weights to learn some new features which are better-suited for the new problem. Since earlier convolutional layers extracts general features and later layers extracts problem-specific features, most transfer learning applications keep the earlier convolutional layers unchanged and retrain only the later layers.

VGG-Net-19 is a deep convolutional neural network consisting of 16 convolutional layers and 3 fully connected layers. Figure 3.5 shows a simplified visual representation of the VGG-Net-19. As described previously, earlier convolutional layers extracts general features like edges, blobs etc. Since these features are common and not specific to any type of image there is no need to retrain earlier layers.

Transfer learning procedure in this work is performed as follows:

- Last two fully connected layers are removed from the VGG-Net-19.
- A fully connected layer with 1024 layers and a fully connected layer with 10 layers are added to the network adjacently. Last layer has made the network to be able to classify between 10 classes of images.
- First 8 convolutional layer has set to keep their weights constant to remain unchanged in retraining phase.
- The network is retrained with a smaller step size than the original training to prevent already learned weights to diverge quickly.

Resulting CNN is used in a same manner as the tracker proposed in [38].

3.3.2 Occlusion Detection

MOSSE [7] tracker has a defined technique to detect occlusion of the target and prevent intensity template to be updated with false matches. On the other hand CNN based tracker [38] has a similar correlation based target matching stage to MOSSE but has no mechanism to detect occlusions. In this thesis an occlusion detection technique similar to those of MOSSE is proposed for CNN based tracker.

CNN based tracker makes its final decision about the location of the target by using the equation 3.43. This final decision gives more weight to the earliest layer. The final decision is more about precise localization of the object and less about differentiating target from other regions. When target appearance rapidly changes, it is expected in this result that there are many similarly high values in non-target regions. Therefore, using PSR metric from MOSSE filter on this weighted correlation output will not yield the expected results because PSR metric tries to calculate how much of an outlier is the peak correlation value, considering the whole image. However, maximum value from the latest layer will remain stable in the fast changing appearance cases and will only decrease when target is not present in the image i.e. occluded. Therefore it is suitable to use weight values which gives the latest layer more weight for PSR calculation. Figure 3.6 shows an example correlation result of the proposed tracker on an example image where an occlusion occurs. It can be seen that maximum correlation value in the latest layer is more close to non-target regions whereas in other layers the output is not much distinct from a non-occlusion case.

Occlusion detection procedure is implemented by firstly checking the equation 3.39 result on the last layer. If the PSR value is below a certain threshold, it is decided that target is not present in the image and thus the maximum correlation result is a false match. In order to update the filter with a non-target region, correlation filter update is disabled for that frame.

Figure 3.7 shows three frames from an experimental run of the proposed method with occlusion detection mechanism is disabled. It can be seen that, during the occlusion, tracker matches with the wrong target and continues to track the wrong target even after the true target reappears. Figure 3.8 shows three frames from an experimental run of the proposed method on the same video sequence with occlusion detection mechanism is enabled. Crossed bounding box in the second frame represents the best target match while being aware that the match is a wrong match. Therefore target appearance model is not updated with the wrong match and tracker successfully reacquires the true target after is reappears.

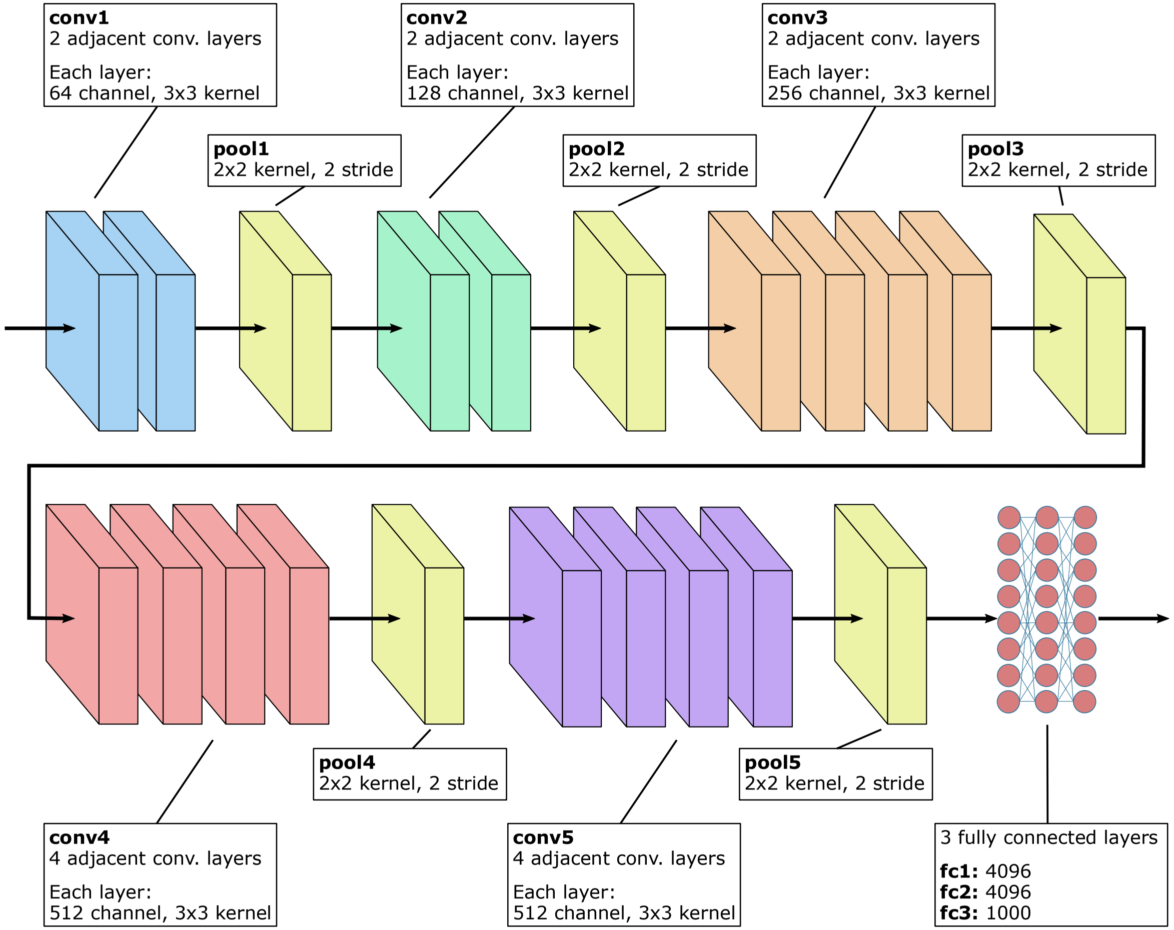
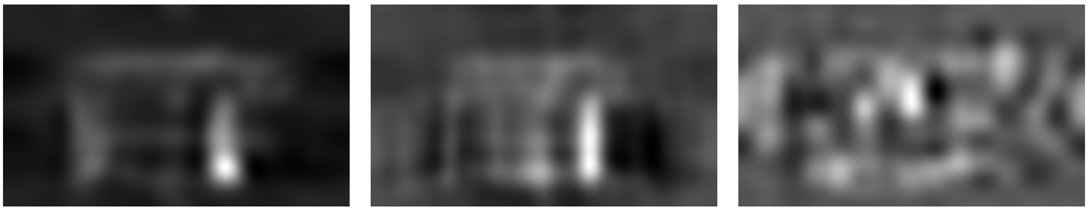
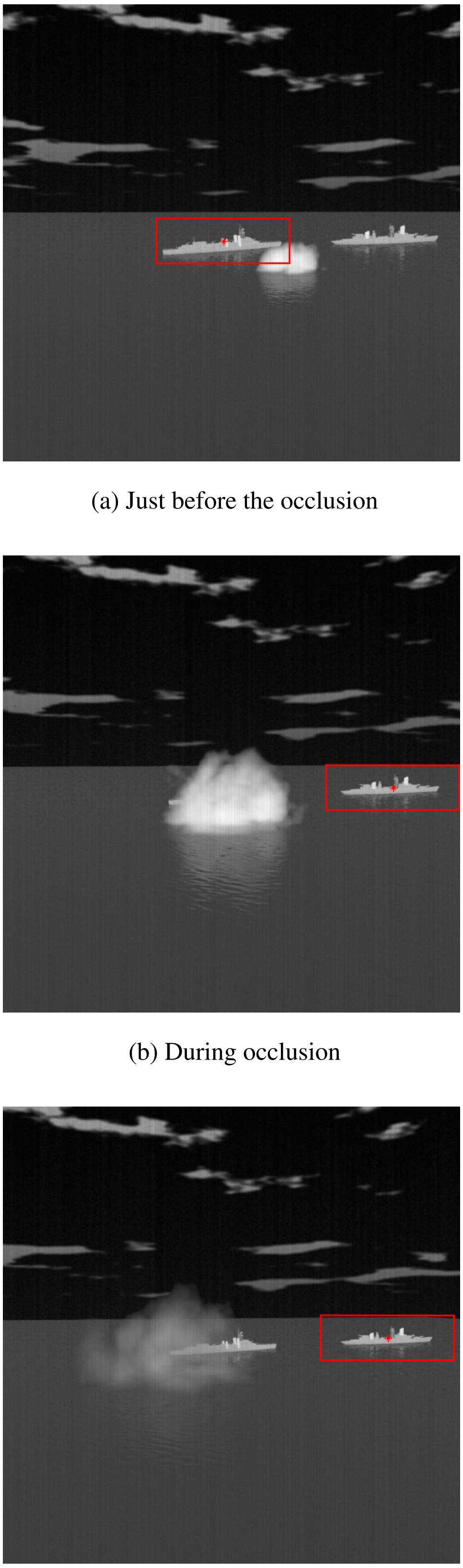


Figure 3.5: VGG-Net-19 layer architecture visualization



(a) Conv3 result (b) Conv4 result (c) Conv5 result

Figure 3.6: Correlation results from different layers.

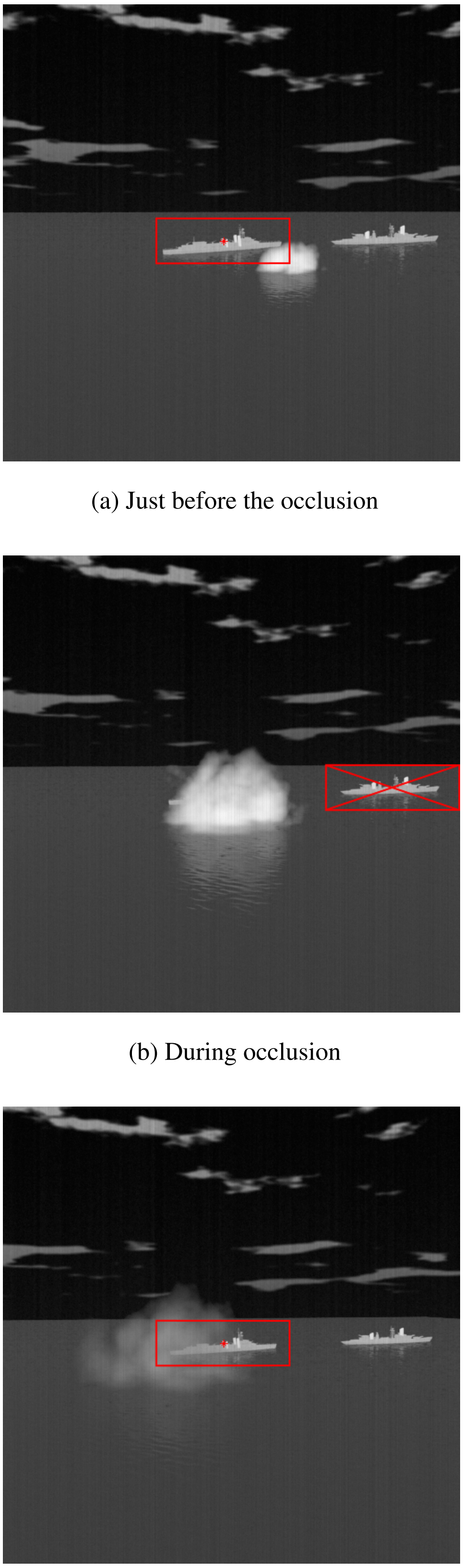


(a) Just before the occlusion

(b) During occlusion

(c) Just after the occlusion

Figure 3.7: Algorithms tracks the wrong target during and after the occlusion



(a) Just before the occlusion

(b) During occlusion

(c) Just after the occlusion

Figure 3.8: Occlusion detection and target reacquisition

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Evaluation of Surface Vessel Tracking Performances

In this work, performances of several visual tracking algorithms are tested. Tests are performed for a situation where image acquiring platform optimally performed small target tracking and the last estimated location of the target is used to initialized the visual target tracking algorithm.

All the algorithms chosen and described in Chapter 3 are tested on annotated video sequences. Video sequences are made up of synthetic IR images of surface watercraft with sea background and annotations are provided as bounding box and centroid positions of the target of interest for each frame. Performances of the target tracking algorithms are assessed by suitable metrics which are described in detail in 4.1.1.

Overall performances are evaluated by the average metric scores of the tracking algorithms. Survival curves for the trackers are also shown in order to better demonstrate the performances of the trackers over different video sequences. Survival curve is a useful tool for assessing tracking performance and used in [47] to compare different tracking algorithms on general object videos.

Tracking algorithms chosen for evaluation and their short names for ease of use is given in Table 4.1.

Prior to the development of our algorithm, the tracker which has the best overall performance was seen to be CNN tracker. It is observed that the tracker uses a convolutional neural network as a feature extractor. The neural network is pre-trained on Image-Net [18] which is an image database comprising visual band images of gen-

Table 4.1: Tracking algorithms chosen for evaluation

Short Name	**Name**
Mean-Shift [14]	Mean Shift Tracking
NCC [37]	Normalized Cross Correlation
MOSSE [7]	Minimum Output Sum of Squared Error Filter
Struck [21]	Structured Output Tracking with Kernels
CNN [38]	Hierarchical Convolutional Features for Visual Tracking
IRS-CNN	Proposed tracker

eral objects. This neural network is too general for out dataset since our dataset only includes IR band images of naval ships. In this thesis a convolutional neural network which is pre-trained on IR band, naval ship images is used in CNN tracker.

4.1.1 Performance Metric

There are many metrics proposed in the literature [47, 2, 41] for the evaluation of visual target tracking performance. These metrics generally require all frames of the video to be annotated such that for every frame there is a bounding box provided for the object being tracked.

F-score [35] is a very popular metric for visual tracking. A single decision is made for each frame of the video sequence regarding the performance of the tracking algorithm. There are three decisions that can be made for a frame:

- True positive: tracker successfully identifies and locates the target.
- False positive: tracker matches with a region which is not the target.
- False negative: tracker fails to find any target in the frame.

While the false negative decision is very clear, there is an overlap criterion to differentiate between true positive and false positive. The overlap criterion, called PASCAL criterion [19], is defined as follows:

$$\frac{|T^i \cap GT^i|}{|T^i \cup GT^i|} \geq 0.5 \tag{4.1}$$

where T^i is estimated target bounding box by the tracker and GT^i is the ground-truth bounding box of the target for the i th frame.

Number of each decision is counted for a video sequence. Number of true positives, false positives and false negatives in a video sequence is represented by n_{tp}, n_{fp}, n_{fn} respectively. F-score is defined as follows [35]:

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4.2}$$

where $precision = n_{tp}/(n_{tp} + n_{fp})$ and $recall = n_{tp}/(n_{tp} + n_{fn})$.

F1-score [31] is a metric similar to F-score but there is not a single decision made for each frame based on thresholded overlap amount. Instead, overlap values is used as is to calculate the metric:

$$F1 = \frac{1}{N_{frames}} \sum_i 2 \cdot \frac{p^i \cdot r^i}{p^i + r^i} \tag{4.3}$$

where $p^i = |T^i \cap GT^i|/|T^i|$ and $r^i = |T^i \cap GT^i|/|GT^i|$.

Overall tracking accuracy (OTA) [3] metric measures tracking performance by using the number o false negatives and false positives (n_{fn}, n_{fp}) in a video sequence. OTA is calculated as follows:

$$OTA = 1 - \frac{n_{fn} + n_{fp}}{\sum_i g^i} \tag{4.4}$$

where g^i is a binary value which indicates if there is a ground truth bounding box available in the frame.

Overall tracking precision (OTP) [32] is a tracking metric which evaluates the tracking performance by using the mean overlap between the tracking bounding boxes (T) and the ground truth bounding boxes (GT). OTP is calculated as follows:

$$OTP = \frac{1}{|M_s|} \sum_{i \in M_s} \frac{|T^i \cap GT^i|}{|T^i \cup GT^i|} \tag{4.5}$$

where M_s is the set of frames where PASCAL overlap criterion is met.

Average Tracking Accuracy (ATA) [31] is a similar metric to OTP, but it uses all the frames for the calculation. ATA is calculated as follows:

$$ATA = \frac{1}{N_{frames}} \sum_{i} \frac{|T^i \cap GT^i|}{|T^i \cup GT^i|} \tag{4.6}$$

where N_{frames} is the number of frames in the whole video sequence.

Deviation [45] is another metric which uses central pixel positions of the tracker and the ground truth. Deviation is calculated as follows:

$$Deviation = 1 - \frac{\sum_{i \in M_s} d(T^i, GT^i)}{|M_s|} \tag{4.7}$$

where $d(T^i, GT^i)$ is the normalized distance between the center pixel of the tracking bounding box and the ground truth bounding box.

In [47], authors performed a comprehensive analysis of the metrics described above. Metrics are tested on 315 different videos to analyse their measurement characteristics. It is observed that all the metrics except deviation has over 0.9 correlation witWh each other in their results. It is inferred that all the metrics except deviation essentially measure the same aspect of tracking performance. Authors then decide to use F-Score for their work for its ease of use over large datasets. Deviation metric is also used, since it measures a very distinct aspect of tracking.

4.1.2 Dataset

Target tracking algorithms are tested on synthetically generated infrared imagery of naval ships. Images consist of 240 video sequences captured at 30 frames per second with a resolution of 512×512. Each video has a duration of 40 seconds. 3D models of the naval ships are obtained from open sources on internet. Synthetic image generator is an industry standard, high radiometric fidelity infrared scene generator. Generated imagery includes a wide variety of scenarios. Scenarios typically consists of a flying image acquiring platform approaching a naval combat ship with different angles and altitude profiles. There is also a second ship always visible alongside the main target in 96 of the 240 videos. Scenarios include infrared countermeasures employed by the ship which are generally infrared flares and infrared emission suppression systems. Scenarios also include challenging situations for target tracking algorithms such as occlusion of the target and rapid change of the appearance due to target or flying platform manoeuvring.

Scenarios are divided into two main categories. First category includes scenarios which has non-manoeuvring target and imaging platform. Both the target and imaging platform has constant velocity. Second category includes scenarios which target or imaging platform or both are manoeuvring. Table 4.2 shows a list of non-manoeuvring scenarios. The list includes different types of ships imaged under several atmospheric conditions. Atmospheric conditions are affected by season of the year and time of the day. All scenarios are run by different relative approach angles and altitudes which are constant in a single run. Relative approach angle set of a scenario includes 5 different angles and starts from 0°and ends at 90°with increments in steps of 22.5°. The altitude set includes 6 different altitudes and starts from 10ft then goes to 100ft and ends at 500ft with increments in steps of 100ft. Therefore, each scenario consists of 30 video sequences with differing approach angles and altitudes. Each run starts with flying platform approaching the target from 10000 meters range and end at 1000 meters away from the target. Figure 4.1 shows a geometric representation of non-manoeuvring scenarios.

Second category of scenarios have the same basic configuration as the first category of scenarios except the target or the imaging platform or both have manoeuvring

Table 4.2: List of non-manoeuvring scenarios.

Id	Target Ship Type	Season And Time	Notes
1	Destroyer	Summer – 07:00	Infrared flare is used by the ship.
2	Battleship	Summer – 14:00	Washdown system is used by the ship.
3	Littoral Combat Ship	Spring – 04:00	-
4	Frigate	Winter – 12:00	Target ship is partially occluded for a while.

motion. Trajectory of the imaging platform is a horizontal sine function with constant altitude. The trajectory deviates a maximum of 250 meters from the straight line trajectory. Unlike the trajectory, orientation of the camera remains constant in a single run and points towards the target. Trajectory of the target is an arc with a radius of 1000 meters. Orientation of the target changes with the trajectory and aligns with the velocity vector. Table 4.3 shows a list of non-manoeuvring scenarios and Figure 4.2 shows a geometric representation manoeuvring scenarios where both the target and imaging platform have a manoeuvring motion.

Figure 4.3, 4.4 and 4.5 are three example images from the dataset. Figure 4.3 is a daylight image of a destroyer in open sea. Figure 4.4 is an image of a battleship acquired at night conditions. Figure 4.5 shows an image of a battleship partially occluded by an infrared flare cloud.

Figure 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12 and 4.13 show example images from each category from the dataset.

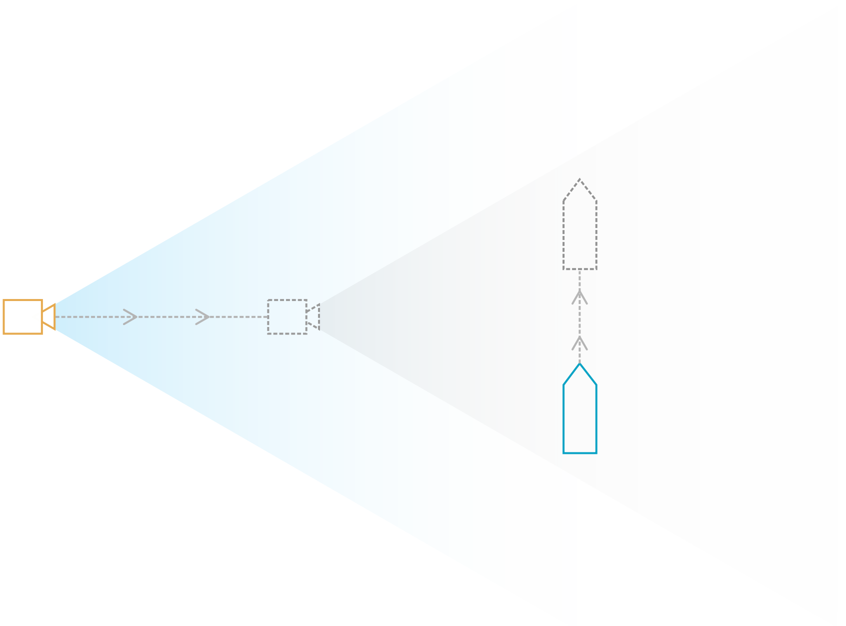


Figure 4.1: Non-manoeuvring scenario

4.1.3 Training Dataset

The ultimate goal of the retraining procedure to increase the object tracking performance. The object tracker need discriminating features to be extracted from image, in order to prevent confusing a different region in the image with the actual target to be tracked. The confused region can be another surface vessel or sea surface or background etc. The dataset classes are chosen according to the situation described above. Classes are defined as follows:

1. Sea background
2. Sky background
3. Land background
4. Infrared countermeasure (Decoy flare)
5. Frigate type surface vessel
6. Destroyer type surface vessel
7. Battleship type surface vessel

Table 4.3: List of manoeuvring scenarios.

Id	Target Ship Type	Season And Time	Notes
5	Destroyer	Summer – 07:00	Ship is manoeuvring. Infrared flare is used by the ship.
6	Battleship	Summer – 14:00	Imaging platform is manoeuvring. Wash-down system is used by the ship.
7	Littoral Combat Ship	Spring – 04:00	Both are manoeuvring.
8	Frigate	Winter – 12:00	Both are manoeuvring. Target ship is partially occluded for a while.

8. Littoral Combat Ship type surface vessel
9. Cruiser type surface vessel
10. Corvette type surface vessel

Figure 4.14, 4.15, 4.16 and 4.17 show example images from the training dataset.

4.2 Experimental Results

Table 4.4 shows overall performance scores of the trackers. Scores are the average score of the tracker on all of the videos in the dataset. It was seen that IRS-CNN out-performs all of the other tracking algorithms.

Table 4.5, 4.6 shows average F-score and Deviation scores of the trackers for different scenario types. Figure 4.18, 4.19 shows the bar graph representation of the same results. Manoeuvring versions of the four categories are shown as bars with darker

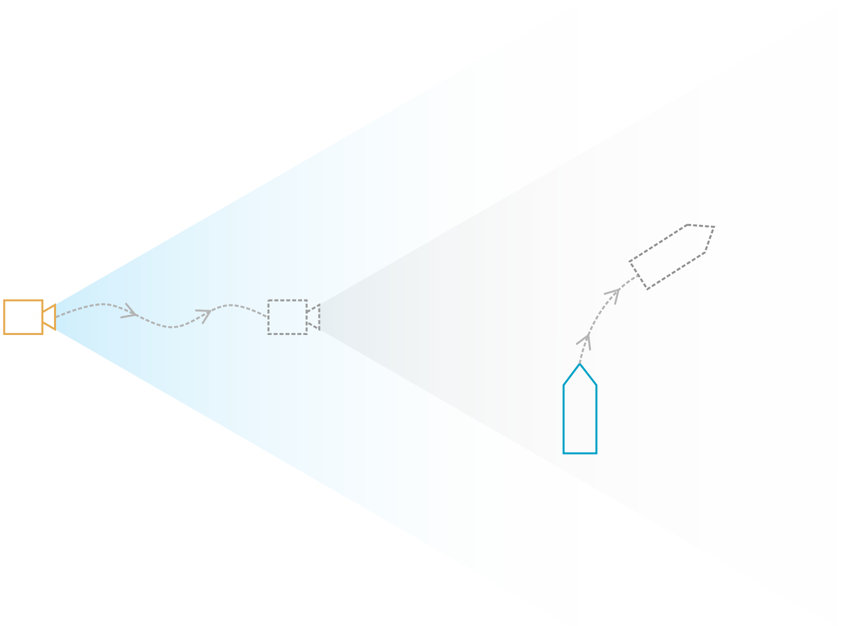


Figure 4.2: Manoeuvring scenario

color in front of the non-manoeuvring versions. Decoy type scenarios are the Scenario 1 and 5, where target uses a bright flare to confuse the trackers. Low intensity type scenarios are 2 and 6 where target uses a cooling system to reduce its visibility, resulting in low intensity values for the target region. Normal type scenarios are 3 and 7 where the target is not actively trying to confuse the tracker. Occlusion type scenarios are 4 and 8 where the target is mostly or completely occluded by infrared flares or by another ship. It should be noted that the performance results of the proposed method for the first 6 categories does not reflect the occlusion detection mechanism since the mechanism is not triggered at all in these categories. On the other hand categories 7 and 8 reflects the compound effect of the both transfer learning and occlusion detection mechanism. While the stand alone effect of the occlusion detection mechanism is not directly measured, it can be inferred from the difference between the normal and occlusion category results since the occlusion and normal categories are the same except the occlusion of the target for a certain amount of the time.

Average scores alone can not represent all information about the performances of the trackers. In order to better demonstrate the performances, survival curves of the scores are given. Survival curves show the scores obtained for each video in the

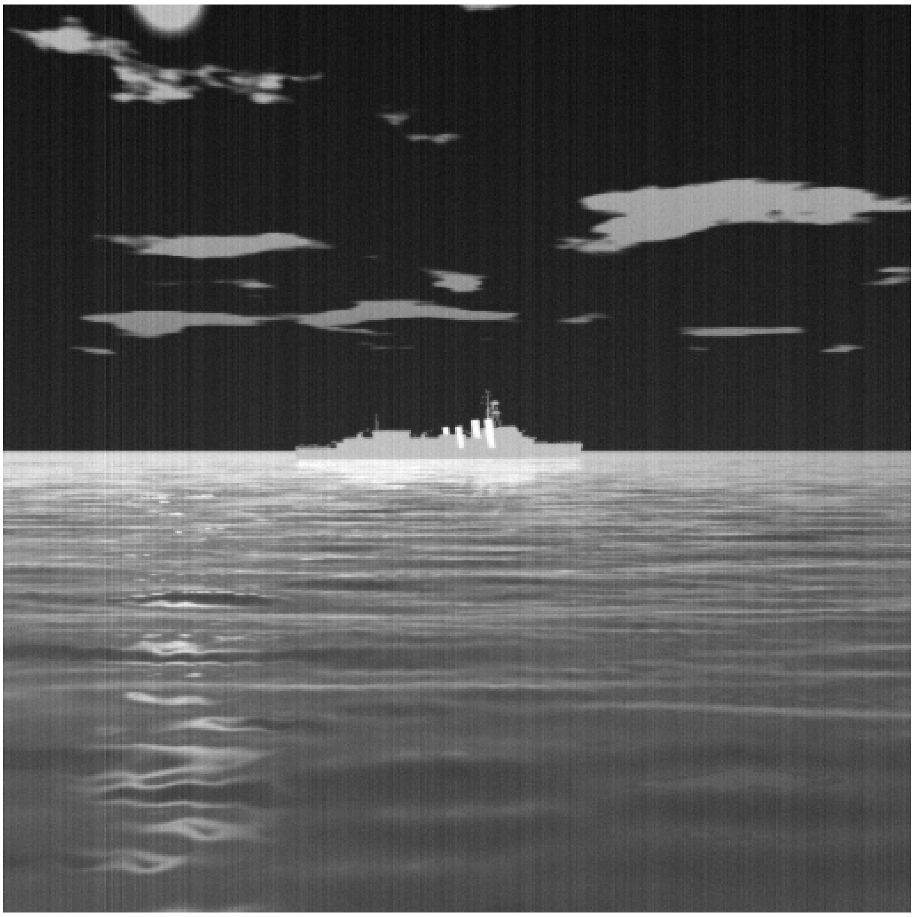


Figure 4.3: Infrared image of a destroyer in open sea

dataset and are prepared by sorting the respective scores of the trackers from 240 videos. First video in the x-axis corresponds to the highest score and the last video corresponds to the lowest score for that tracker. Note that this means order of the videos is different for each tracker. Figure 4.20 and 4.20 shows the survival curves of the trackers for F-score and deviation score respectively.

Results show a distinct nature between the F-scores and Deviation scores. Therefore their results are discussed separately. Following comments concerning the tracking performances are about the inferences made from the F-scores. Deviation scores are discussed in 4.2.1.

Lower overall performance of Mean-Shift tracker can be easily seen from the results. This result can be attributed the target representation model of the algorithm. Since Mean-Shift uses an histogram based target representation, tracker can easily be con-

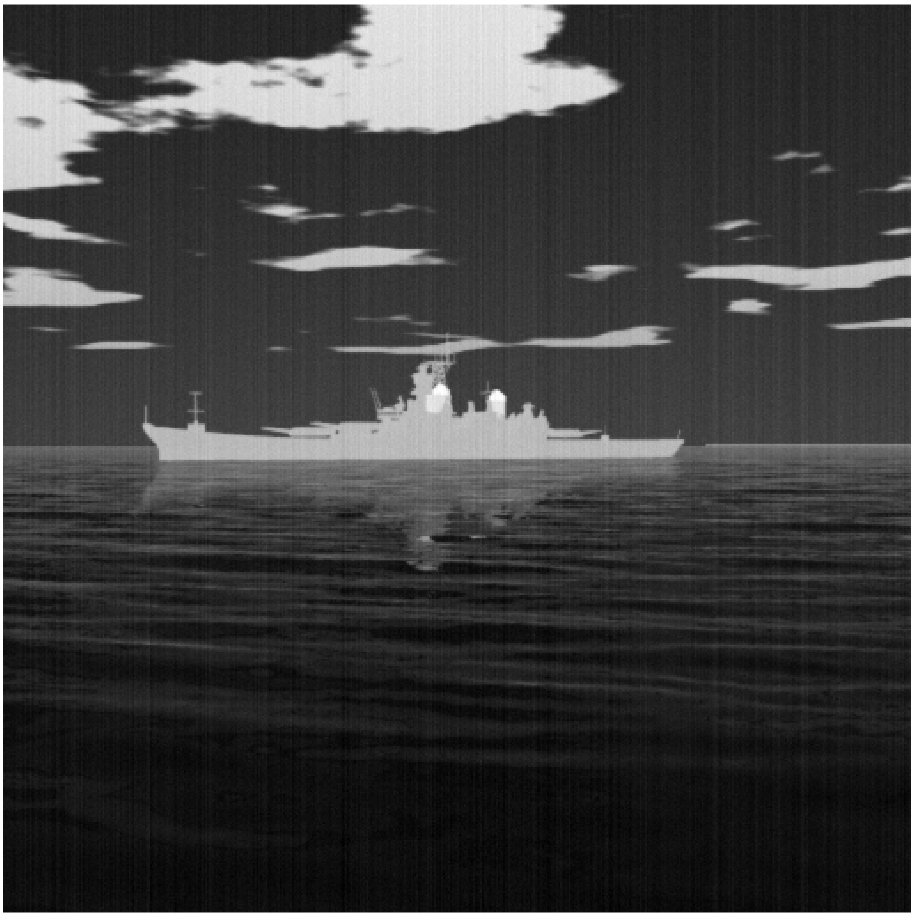


Figure 4.4: Infrared image of a battleship in open sea

fused with a region with a similar intensity distribution as the target. This situation especially occurs in low intensity category scenarios of our dataset because target cools itself down with sea water which is also the main background clutter in our dataset, thus makes the target region histogram to be very similar to background. In infrared decoy type scenarios, decoy is generally appear brighter with a distinct histogram from the target. This results in a relatively better overall performance in decoy type scenarios compared to low intensity type scenarios. Furthermore, upon closer inspection it was seen that tracker failed in some cases of decoy type scenarios where decoy is not as bright and has similar intensity to the target. Mean-Shift has similar overall performance values between manoeuvring and non-manoeuvring type scenarios because intensity distribution does not change dramatically with the changing appearance of the target.

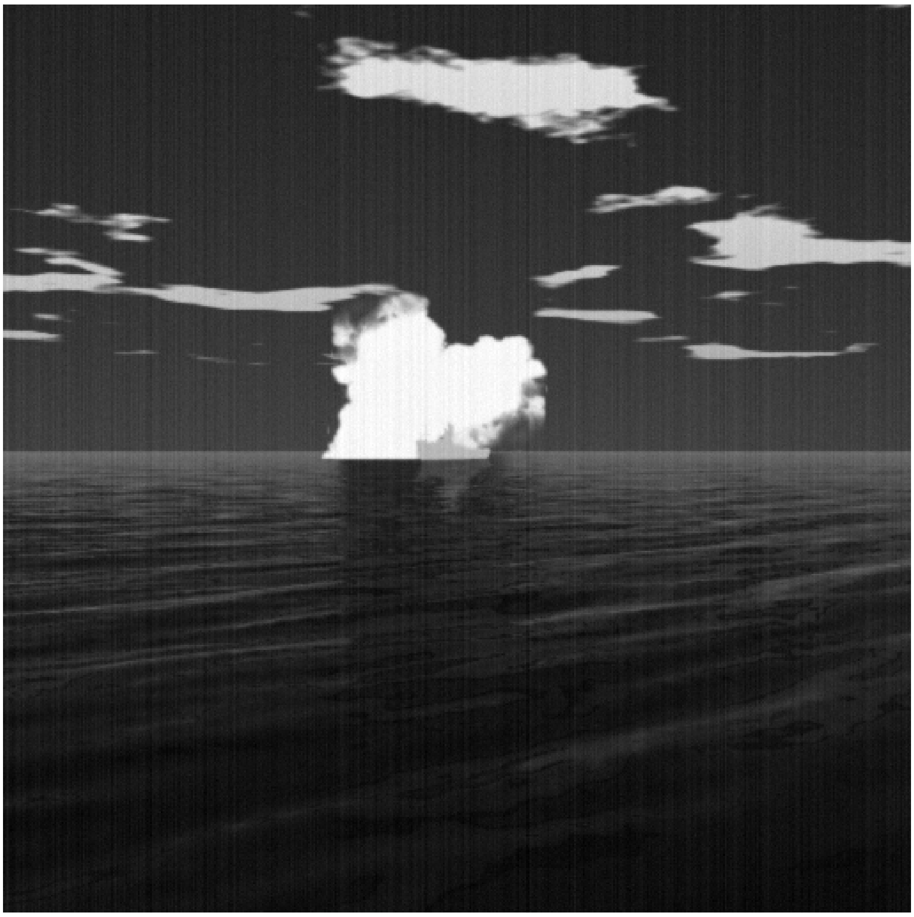


Figure 4.5: Infrared image of a destroyer partially occluded by infrared flare

Normalized Cross Correlation (NCC) tracker has the second lowest overall performance but still performs noticeably close to top-performing trackers. It is noteworthy that as a tracker which is developed in the 90's and has a simplistic approach to tracking problem, NCC has such reasonable performance. It is observed that NCC performed worse in the manoeuvring type scenarios. This can be explained by that intensity template of NCC which is simply the normalized image of the target which holds little semantic or discriminative information. Such type of target template is not robust against the rapid changing appearance of the target as it tries to match nearly exact appearance as the previous frames.

Intensity template is also the target representation for the MOSSE tracker. However, results show that performance of MOSSE in manoeuvring type scenarios is only slightly worse and difference is not dramatic as in the case of NCC. This can be ex-

Table 4.4: F-scores and Deviation scores of the trackers

Tracker	F-Score	Deviation Score
Mean-Shift	0.380	0.716
NCC	0.532	0.824
MOSSE	0.592	0.842
Struck	0.625	0.870
CNN	0.673	0.854
IRS-CNN	0.698	0.856

plained by the template generation technique of the MOSSE in the new frame. The filter generated for the new frame correlates positively with the target region but that has no correlation with the other regions in the frame. This means that unlike NCC, MOSSE intensity template has discriminative properties. Thus, rapid changing of appearance affects the correlation result at the location of target negatively but this does not generally confuse the tracker since correlation results at other locations was already much lower. Results show that discriminative template not only helps with the rapid appearance change but also increase the overall performance in every category. It can also be seen that occlusion detection mechanics of the MOSSE greatly helps the occlusion type scenario performance.

Struck has very favourable performances in all categories. The generalization ability of the underlying SVM framework enables the tracker to be robust against fast changing appearance. This results in a very little performance difference between manoeuvring and non-manoeuvring type scenarios. In the occlusion type scenarios Struck performs slightly worse compared to other types of scenarios but the difference is negligible. The generalization ability of the tracker, again, helps with the cases where target is occluded for a short amount of time. In those cases Struck successfully reacquires the target even if the target appearance is changed moderately.

Table 4.5: F-scores of the trackers for different categories

Scenario Difficulty	Mean-Shift	NCC	MOSSE	Struck	CNN	IRS-CNN
Decoy (NM)	0.362	0.556	0.582	0.612	0.668	0.687
Decoy (M)	0.348	0.504	0.565	0.606	0.664	0.682
Low Intensity (NM)	0.331	0.558	0.583	0.615	0.670	0.688
Low Intensity (M)	0.321	0.505	0.565	0.605	0.662	0.681
Normal (NM)	0.460	0.602	0.630	0.662	0.725	0.727
Normal (M)	0.447	0.550	0.613	0.656	0.718	0.724
Occlusion (NM)	0.390	0.514	0.606	0.624	0.644	0.702
Occlusion (M)	0.381	0.467	0.590	0.620	0.634	0.694

The outstanding performance of CNN based tracker is also apparent in the results. This method uses correlation filters in a similar manner to those of MOSSE. But, unlike MOSSE, this tracker employs multiple correlation filters working on different features extracted from the image. These features are not hand picked features but rather automatically generated as a result of a CNN training. Since the goal of the tracking was to classify different kinds of objects, resulting features hold a discriminative quality. Features from earlier layers of the CNN are capable of extracting semantic features which produce different correlation results for different objects even when the intensity values of the objects is very similar. This helps tracker to not confuse the target with other objects present in the scene. Therefore, overall significant performance increase from MOSSE can be explained by the advantage of multi-channel filters working on features optimized for classification. While being in the second place in occlusion category, CNN based tracker has the lowest of its scores in this category. Although the semantic awareness of the tracker compensates substantially for the lack of an occlusion detection mechanism, this shortcoming is still perceivable to a small extent.

Proposed method exhibits varied amounts of increase in performance compared to CNN based tracker. This improvement can be explained by the concentrated discriminative ability of the underlying tracker. Original CNN based tracker has a feature

Table 4.6: Deviation scores of the trackers for different categories

Scenario Difficulty	**Mean-Shift**	**NCC**	**MOSSE**	**Struck**	**CNN**	**IRS-CNN**
Decoy (NM)	0.719	0.862	0.848	0.866	0.854	0.853
Decoy (M)	0.692	0.787	0.821	0.861	0.846	0.844
Low Intensity (NM)	0.704	0.864	0.850	0.868	0.856	0.852
Low Intensity (M)	0.675	0.787	0.822	0.862	0.849	0.846
Normal (NM)	0.765	0.878	0.865	0.889	0.875	0.872
Normal (M)	0.735	0.803	0.838	0.878	0.866	0.862
Occlusion (NM)	0.733	0.846	0.860	0.875	0.847	0.864
Occlusion (M)	0.706	0.766	0.832	0.862	0.839	0.854

extractor tailored for classifying RGB colorspace images of 1000 different object types. Our problem requires feature extractors for grayscale colorspace images of naval combat environment which are under-represented in the training data of the original CNN based tracker. Most of the time, different objects in a scene from out dataset falls into the same class of the general object dataset. This implies that different objects in our dataset produces similar semantic results with each other. This effect compromises target tracking performance by causing the tracker to confuse a non-target region with our target. Our tracker improves this shortcoming by employing a transfer learning procedure with a dataset consisting of different types of surface vessels, common background types and countermeasures. Second improvement can be observed on the occlusion category scores. Even though occlusion category scores are still the worst scores of the tracker, difference with other categories is negligible and it is a higher score than the occlusion score of the CNN based tracker. This can be interpreted as a result of the proposed occlusion detection mechanism of our tracker.

It is also observed that none of the trackers has noticeable performance degradation in decoy type scenarios. It can be said that, as long as it is not occluding the target, infrared decoys has no noticeable effect on the visual tracking performance of the modern imaging infrared sensors.

4.2.1 Localization Accuracy

Compared to F-scores, deviation scores have less information about the ability of the tracker to find the correct target. Deviation scores basically represent localization accuracy. Histogram based representation of the Mean-Shift tracker causes, again, the tracker to have the lowest scores. Small translations of the tracking window does not change the histogram dramatically and this effect causes Mean-Shift tracker to achieve a less sensitive localization performance compared to other trackers. On the other hand, there is no significant difference between the performances of the other trackers. This result is not surprising for NCC, CNN and proposed tracker since these trackers all have an intensity template based target representation and correlation based matching stages. Correlation of an intensity template is very sensitive to translation because location of the pixels must fit almost exactly to achieve a high correlation result. Figure 4.22 shows a comparison between the cost function of the Mean-shift tracker (Bhattacharyya coefficient) and the correlation result of the NCC on an example image from our dataset. Figure actually shows Bhattacharyya coefficient subtracted from 1 for comparison purposes since it is a cost function and the target located at the minimum value. It can be seen that correlation result of the NCC tracker produces a strong peak and the cost function of the Mean-shift tracker has a very dull peak at the target location which is marked with a small red circle in both figures.

On the other hand, Struck tracker trains a classifier which directly maps images to translations. Therefore, core filter update optimization procedure primarily deals with minimizing the location error. This explains the outstanding localization performance of the Struck tracker.

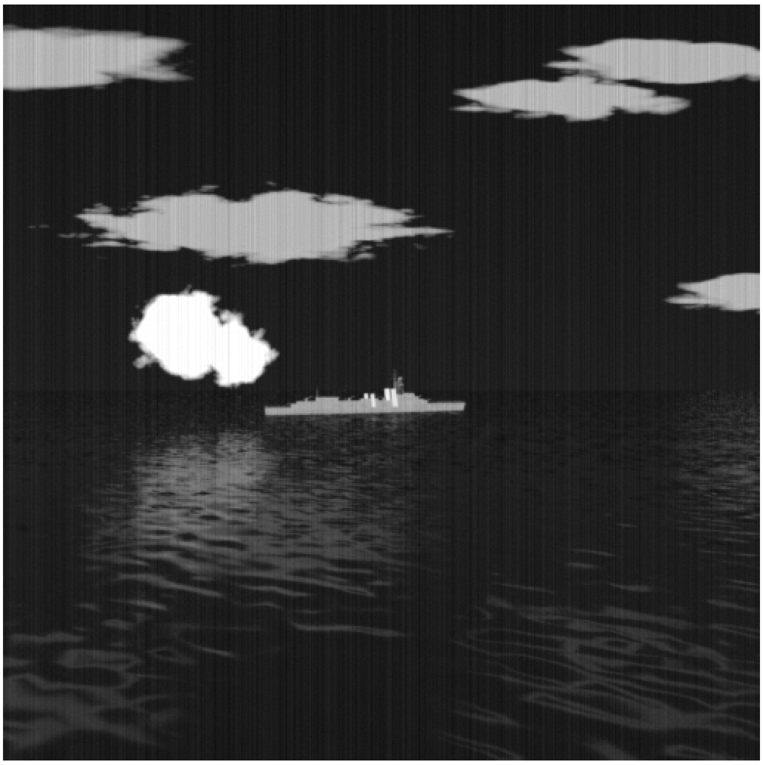


Figure 4.6: Example image from the scenario category 1

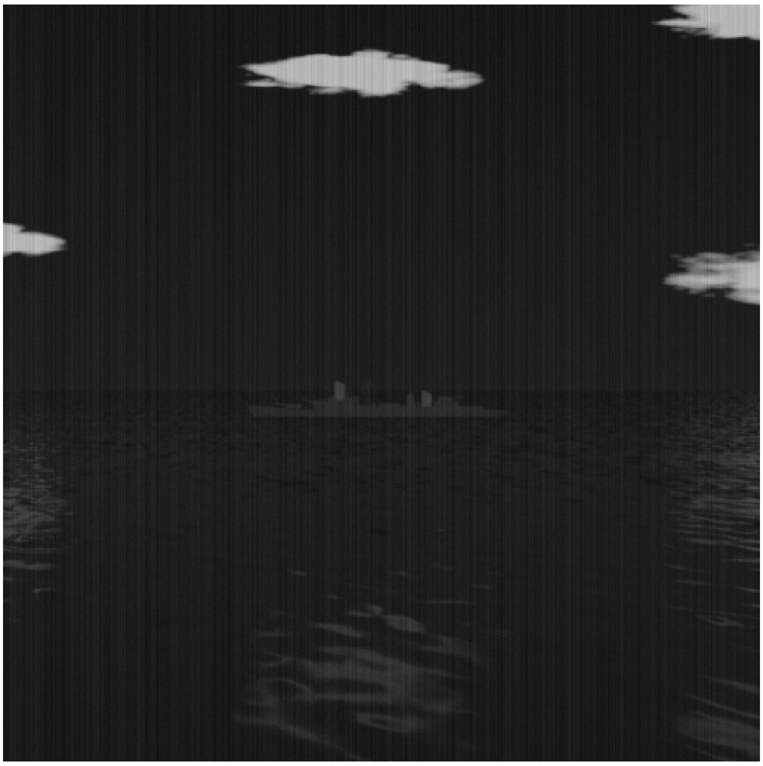


Figure 4.7: Example image from the scenario category 2

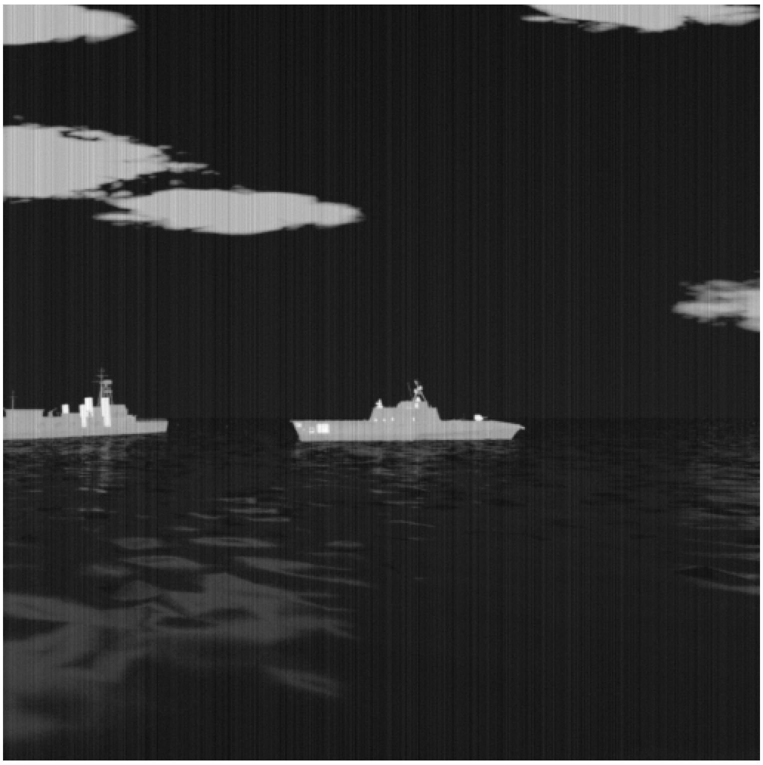


Figure 4.8: Example image from the scenario category 3

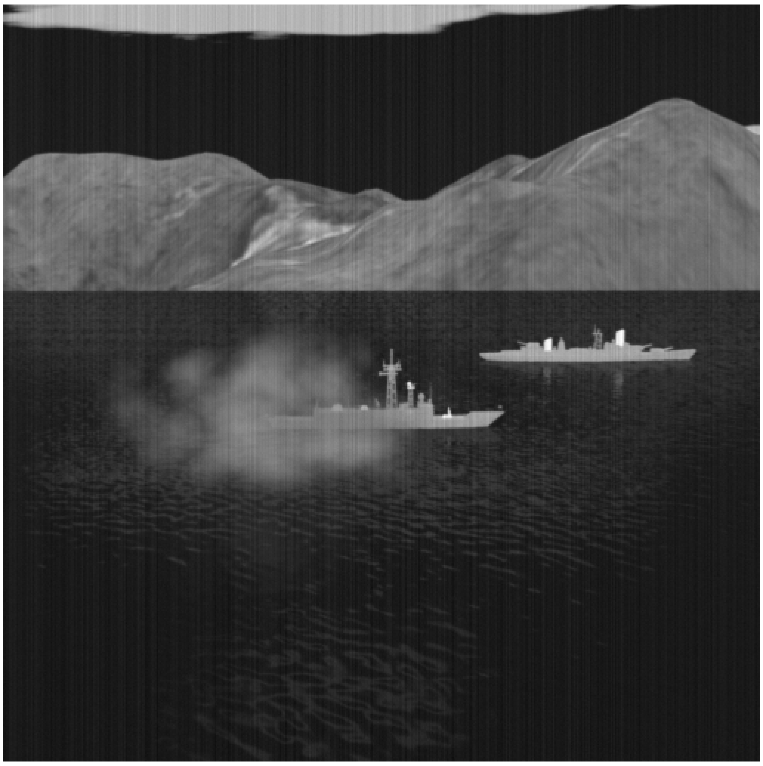


Figure 4.9: Example image from the scenario category 4

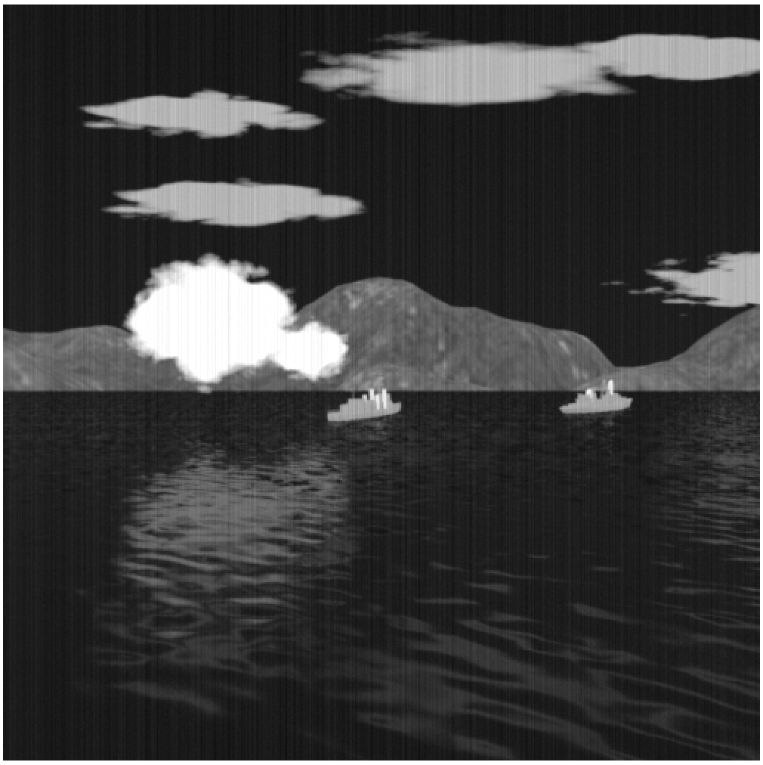


Figure 4.10: Example image from the scenario category 5

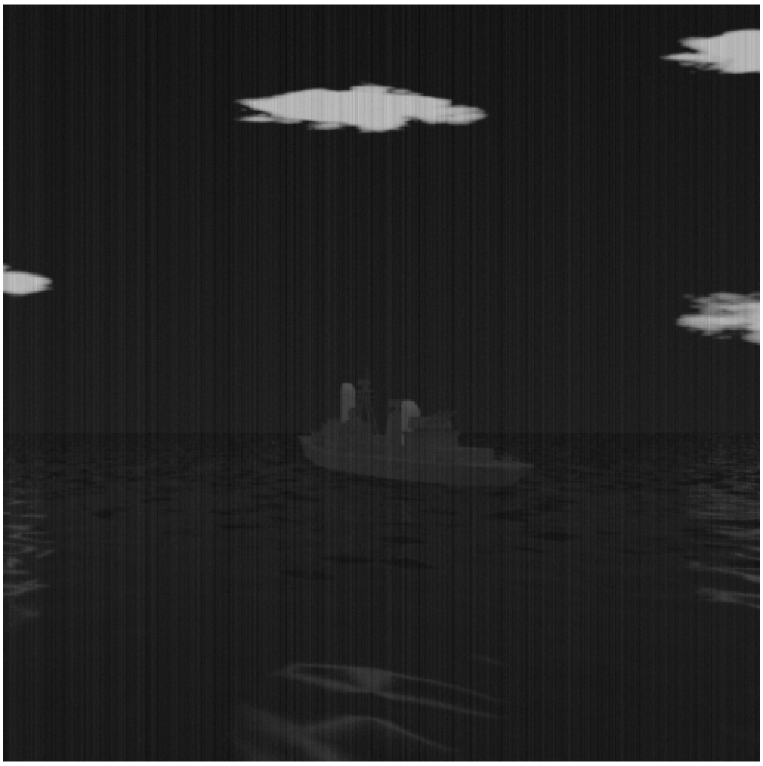


Figure 4.11: Example image from the scenario category 6

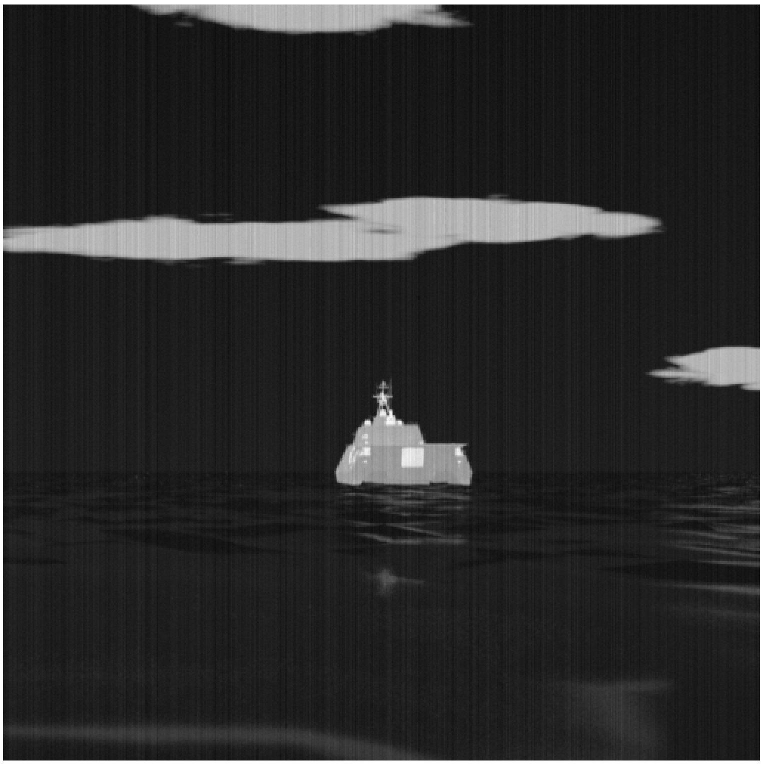


Figure 4.12: Example image from the scenario category 7

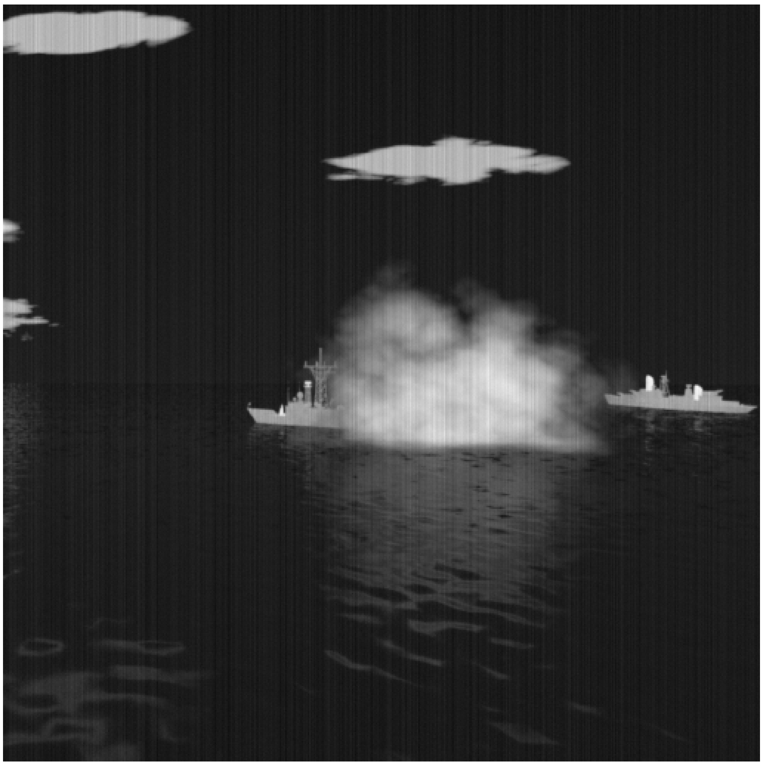


Figure 4.13: Example image from the scenario category 8

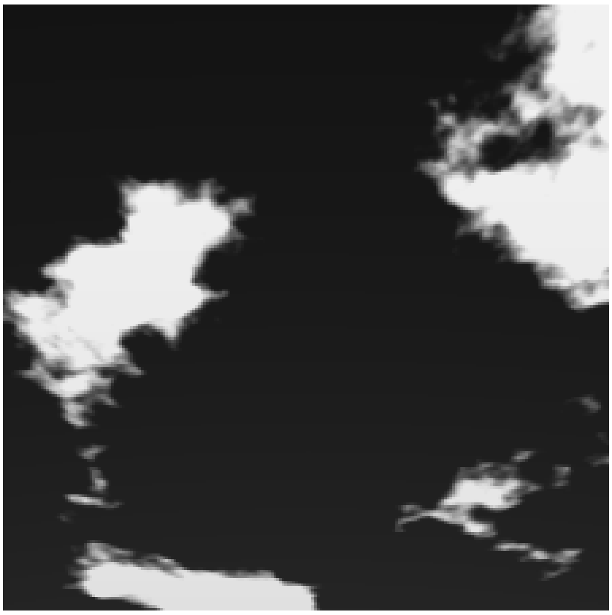


Figure 4.14: Example image from Sky background class

Figure 4.15: Example image from Sea background class

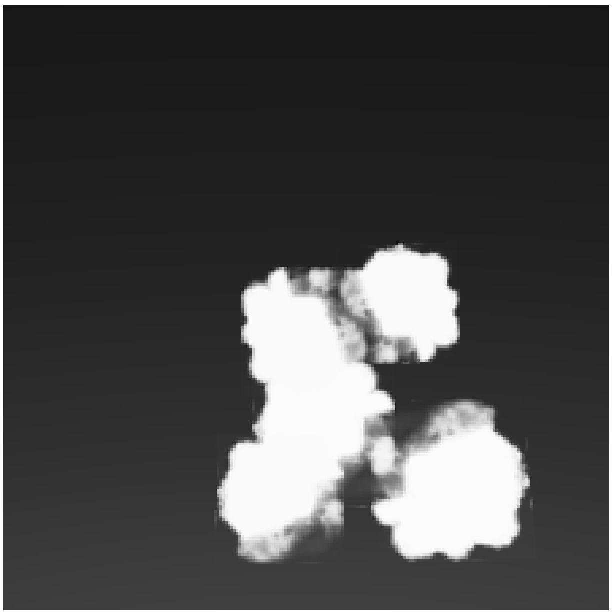


Figure 4.16: Example image from Infrared countermeasure class

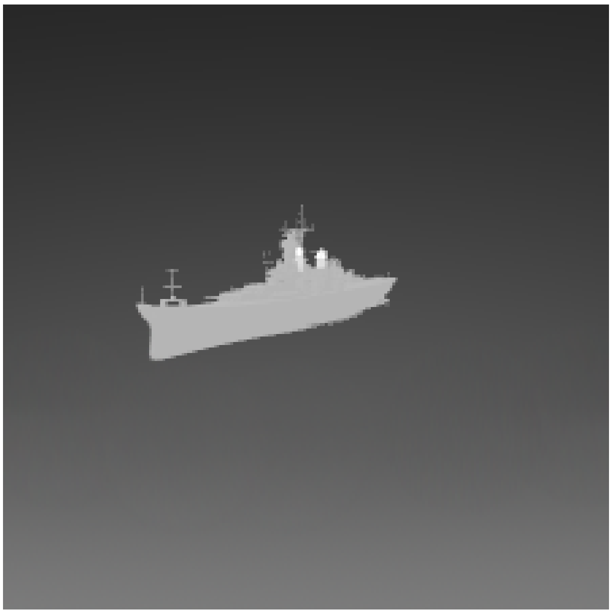


Figure 4.17: Example image from Battleship class

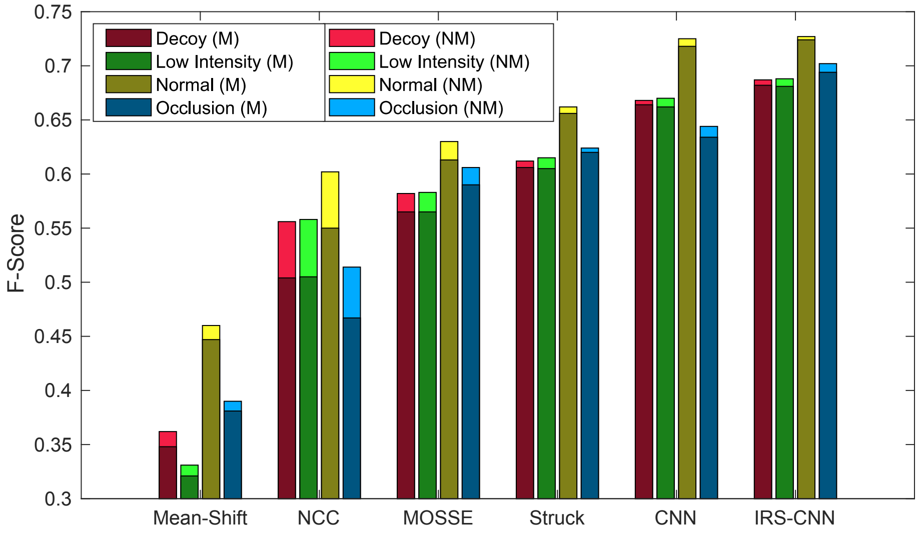


Figure 4.18: F-score bar graphs of the trackers for different categories

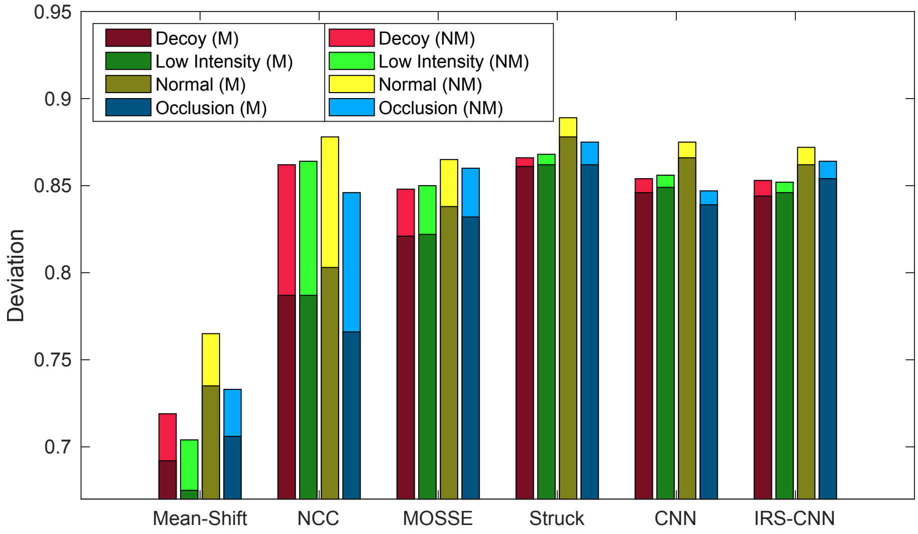


Figure 4.19: Deviation bar graphs of the trackers for different categories

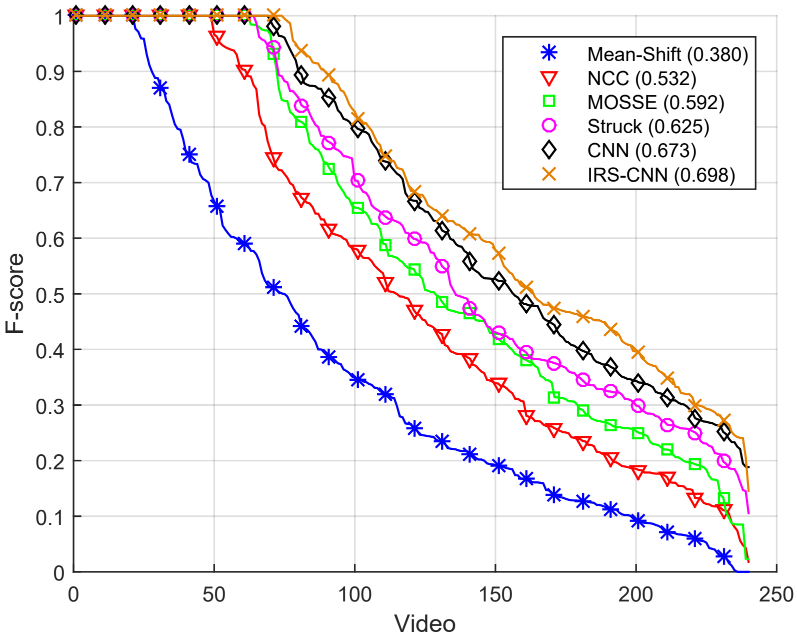


Figure 4.20: Survival curves of the trackers (F-scores)

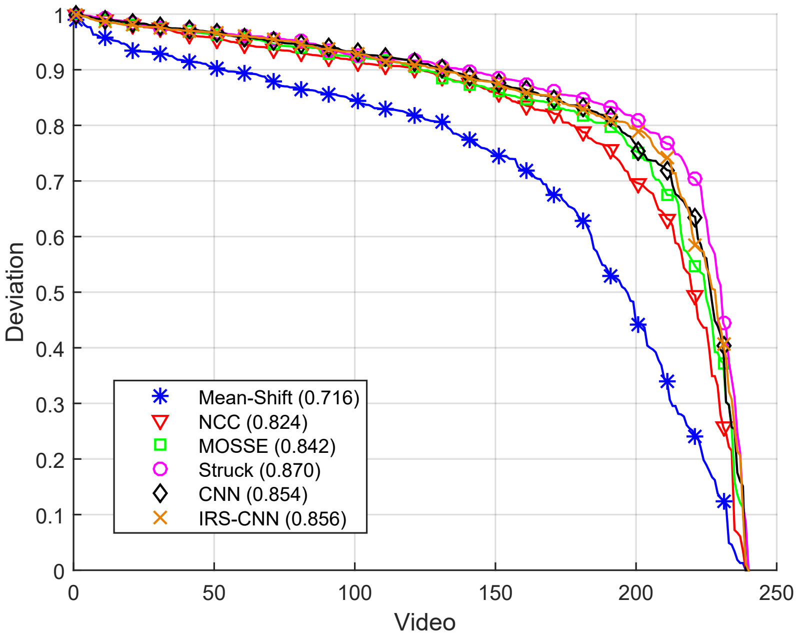
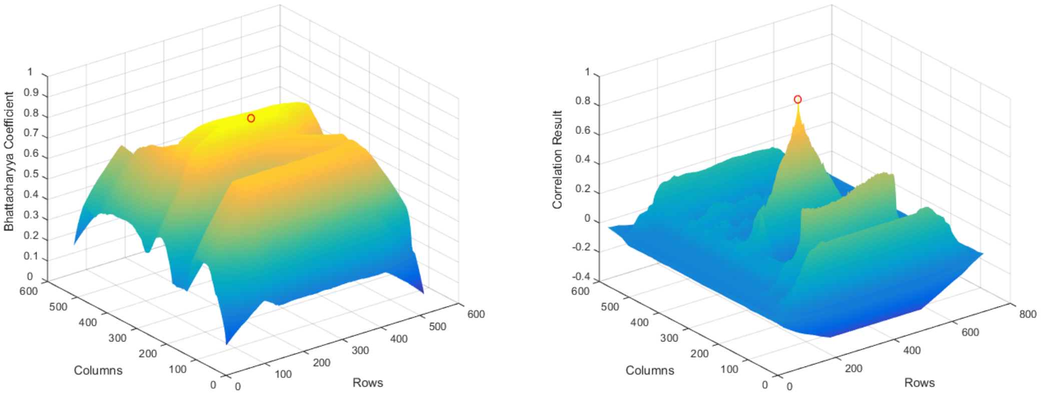


Figure 4.21: Survival curves of the trackers (Deviation scores)



(a) Mean-Shift Bhattacharyya coefficient

(b) NCC result

Figure 4.22: Target matching stage outputs of (a) Mean-Shift and (b) NCC trackers

CHAPTER 5

CONCLUSION

Tracking the locations of enemy watercraft is a highly crucial task in a naval combat environment. Generally first choice for a sensor is radar, but radars are active systems and emits RF signals which can reveal our presence and location to the enemy. Therefore, infrared imaging systems are employed for tracking enemy watercraft.

There is substantial amount of work in literature and many different models of appearance are used to differentiate the object of interest from other objects or regions in the new image. In some works, intensity distribution of the object is utilized for locating the object. Certain algorithms use the intensities as template and use correlation to find the object. SVM and CNN based trackers intend to classify the regions of the image to differentiate the object from its surroundings.

The main algorithm which we focus on is a CNN and template based tracker hybrid. This technique offers fast computation times with its use of correlation instead of a classification task and benefits from the feature extraction capabilities of deep convolutional neural networks.

Distinguished object tracking algorithms are either implemented by ourselves or obtained from the website of the developer. It was seen in our tests that CNN based tracker had superior performance to the other trackers.

The main motivation of this thesis is tracking combat ships only. Like any other tracker, CNN based tracker is too general for our needs, but it also has the advantage that can allow us to make the algorithm to track specific object types. These concentration on specific object types can reduce the general object tracking performance of the tracker, but it increases the tracking performance for our dataset.

Underlying CNN feature extractor of the tracker has been retrained in the later layers which are the problem specific feature extraction layers. The CNN is retrained to be able to classify between different background and different types of combat ships. This manner of retraining lead the network to learn discriminating convolutional features between these object types. Therefore every convolutional layer generated different levels of response for different objects. This ensured tracker to not confuse the object of interest with any other region in the image, thus increased the tracking performance.

Tracking performances are evaluated with prevalent performance metrics found in the literature. The algorithms are tested on a specifically tailored dataset. Dataset comprises of infrared images of combat ships under different atmospheric situations. Bounding-box annotations ships are also generated to be used as ground-truth data in the performance tests.

Performances are presented by means of average metric scores of the trackers and survival curves of the trackers over the whole dataset. Survival curve is a useful tool to represent the performance of the tracker in detail by showing the score distribution of the tracker over different video sequences.

It was seen that proposed algorithm had superior performance compared to the other trackers. This result shows that convolutional features can be adapted to different scenarios to increase tracking performance. In the future, proposed algorithm can be experimented with different tracking problem such as tracking aircraft or surveillance of a specific type of object.

REFERENCES

[1] Y. Barniv. Dynamic programming solution for detecting dim moving targets. *IEEE Transactions on Aerospace and Electronic Systems*, (1):144–156, 1985.

[2] F. Bashir and F. Porikli. Performance evaluation of object detection and tracking systems. In *Proceedings 9th IEEE International Workshop on PETS*, pages 7–14, 2006.

[3] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, 2008:1, 2008.

[4] D. Beymer and K. Konolige. Real-time tracking of multiple people using continuous detection. In *IEEE Frame Rate Workshop*, page 53. Citeseer, 1999.

[5] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 232–237. IEEE, 1998.

[6] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *European conference on computer vision*, pages 2–15. Springer, 2008.

[7] D. S. Bolme, J. R. Beveridge, B. Draper, Y. M. Lui, et al. Visual object tracking using adaptive correlation filters. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2544–2550. IEEE, 2010.

[8] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 1998.

[9] K. Briechle and U. D. Hanebeck. Template matching using fast normalized cross correlation. In *Aerospace/Defense Sensing, Simulation, and Controls*, pages 95–102. International Society for Optics and Photonics, 2001.

[10] T. J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE transactions on pattern analysis and machine intelligence*, (1):90–99, 1986.

[11] A. Çakıroğlu. Tracking variable number of targets with joint probabilistic data association filter. In *2016 24th Signal Processing and Communication Application Conference (SIU)*, pages 2017–2020. IEEE, 2016.

[12] Y. Chen, Y. Rui, and T. S. Huang. Jpdaf based hmm for real-time contour tracking. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–543. IEEE, 2001.

[13] J. U. Cho, S. H. Jin, X. Dai Pham, J. W. Jeon, J. E. Byun, and H. Kang. A real-time object tracking system using a particle filter. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2822–2827. IEEE, 2006.

[14] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564–577, 2003.

[15] I. J. Cox and S. L. Hingorani. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 18(2):138–150, 1996.

[16] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.

[17] S. J. Davey, M. G. Rutten, and B. Cheung. A comparison of detection performance for several track-before-detect algorithms. *EURASIP Journal on Advances in Signal Processing*, 2008:41, 2008.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[19] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[20] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Multi-target tracking using joint probabilistic data association. In *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*, pages 807–812. IEEE, 1980.

[21] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 263–270. IEEE, 2011.

[22] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang. Visual tracking via locality sensitive histograms. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2427–2434. IEEE, 2013.

[23] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, pages 749–765. Springer, 2016.

[24] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Computer Vision–ECCV 2012*, pages 702–715. Springer, 2012.

[25] C. Hue, J.-P. Le Cadre, and P. Pérez. Tracking multiple objects with particle filtering. *IEEE transactions on aerospace and electronic systems*, 38(3):791–812, 2002.

[26] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Computer Vision—ECCV'96*, pages 343–356. Springer, 1996.

[27] M. Isard and A. Blake. Condensation—conditional density propagation for visual tracking. *International journal of computer vision*, 29(1):5–28, 1998.

[28] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1296–1311, 2003.

[29] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2012.

[30] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[31] B. Karasulu and S. Korukoglu. A software for performance evaluation and comparison of people detection and tracking methods in video processing. *Multimedia Tools and Applications*, 55(3):677–723, 2011.

[32] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):319–336, 2009.

[33] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, et al. The visual object tracking vot2015 challenge results. http://www.votchallenge.net/vot2015/download/vot_2015_paper.pdf. Accessed: 2016-01-13.

[34] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin, G. Nebehay, T. Vojíř, G. Fernandez, A. Lukežič, A. Dimitriev, et al. The visual object tracking vot2014 challenge results. In *Computer Vision-ECCV 2014 Workshops*, pages 191–217. Springer, 2014.

[35] J. Kwon and K. M. Lee. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1208–1215. IEEE, 2009.

[36] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer, 1999.

[37] J. Lewis. Fast normalized cross-correlation. In *Vision interface*, volume 10, pages 120–123, 1995.

[38] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 3074–3082, 2015.

[39] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):810–815, 2004.

[40] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. *arXiv preprint arXiv:1510.07945*, 2015.

[41] A. T. Nghiem, F. Bremond, M. Thonnat, and V. Valentin. Etiseo, performance evaluation for video surveillance systems. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 476–481. IEEE, 2007.

[42] I. S. Reed, R. M. Gagliardi, and L. B. Stotts. Optical moving target detection with 3-d matched filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 24(4):327–336, 1988.

[43] D. Reid. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, 24(6):843–854, 1979.

[44] R. Rosales and S. Sclaroff. 3d trajectory recovery for tracking multiple objects and trajectory guided recognition of actions. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 117–123. IEEE, 1999.

[45] A. Sanin, C. Sanderson, and B. C. Lovell. Shadow detection: A survey and comparative evaluation of recent methods. *Pattern recognition*, 45(4):1684–1695, 2012.

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

[47] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, July 2014.

[48] R. L. Streit and T. E. Luginbuhl. Maximum likelihood method for probabilistic multihypothesis tracking. In *SPIE's International Symposium on Optical Engineering and Photonics in Aerospace Sensing*, pages 394–405. International Society for Optics and Photonics, 1994.

[49] N. Wang and D.-Y. Yeung. Learning a deep compact image representation for visual tracking. In *Advances in neural information processing systems*, pages 809–817, 2013.

[50] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

[51] A. Yilmaz, X. Li, and M. Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(11):1531–1536, 2004.

[52] F. Zhang, C. Li, and L. Shi. Detecting and tracking dim moving point target in ir image sequence. *Infrared Physics & Technology*, 46(4):323–328, 2005.